

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ ГАГАРИНА Ю.А.»

Кафедра «Естественные и математические науки»

Специальность ПИНЖ — Программная инженерия

Проект базы данных к информационной системе
«Магазин программных продуктов»

Выполнил: студент X курса
учебной группы XXXX
очной формы обучения
XXXXXXX,

Проверил: преподаватель кафедры
ЕМН XXXXXX

Энгельс 2025

СОДЕРЖАНИЕ

1 База данных	3
1.1 Анализ предметной области «Программные продукты»	3
1.2 Модель сущность-связь	9
1.3 Дата-логическое (логическое) проектирование базы данных	10
1.4 Физическое моделирование базы данных	12
2 Функциональные возможности	21
2.1 Запросы на редактирование	21
2.1.1 Запрос на добавление записи в таблицу	21
2.1.2 Запрос на изменение данных таблицы	21
2.1.3 Запрос на удаление данных из таблицы	22
2.2 Запросы на выборку данных	22
2.2.1 Запрос на фильтрацию данных из таблицы	22
2.2.2 Запрос с использованием агрегатных функций	23
2.2.3 Запрос с объединением нескольких таблиц (JOIN)	24
2.2.4 Запрос на проверку уникальности данных	25
2.3 Триггеры	25
2.4 Процедуры	27
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	29

1 База данных

1.1 Анализ предметной области «Программные продукты»

Областью автоматизации является процесс реализации цифровых лицензий на программное обеспечение (ПО) и управление клиентской базой в рамках информационной системы.

В настоящее время на рынке цифровых товаров наблюдается потребность в централизованных платформах, обеспечивающих прозрачный процесс приобретения лицензионных ключей, их активации и контроля сроков действия. Разрабатываемая система предназначена для автоматизации работы администратора магазина по управлению каталогом и заказами, а также для предоставления клиентам удобного интерфейса выбора и покупки ПО.

Основной акцент в рамках данной работы сделан на модулях «Каталог товаров», «Управление заказами» и «Система лицензирования».

Нормативные документы

1. Гражданский кодекс Российской Федерации (часть четвертая) [Электронный ресурс] : федер. закон от 18.12.2006 № 230-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_64629/. – (Дата обращения: 14.10.2025).

Регламентирует правовую охрану программ для ЭВМ и баз данных, а также порядок заключения лицензионных договоров, что обосновывает наличие таблицы «Лицензии» и полей с типами ключей в БД.

2. Российская Федерация. Законы. О защите прав потребителей [Электронный ресурс] : закон от 07.02.1992 № 2300-1 (ред. от 25.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_305/. – (Дата обращения: 14.10.2025).

Обосновывает необходимость предоставления полной информации о товаре (таблица «Товары», поля «Описание», «Цена») и правила возврата/обмена цифровых товаров.

3. Об утверждении Правил продажи товаров по договору розничной купли-продажи, перечня товаров длительного пользования, на которые не распространяется требование покупателя о безвозмездном предоставлении ему на период ремонта или замены аналогичного товара, и перечня непродовольственных товаров надлежащего качества, не подлежащих обмену, а также о внесении изменений в некоторые акты Правительства Российской Федерации [Электронный ресурс] : постановление Правительства Рос. Федерации от 31.12.2020 № 2463 (ред. от 01.04.2024) // Гарант. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_373622/. – (Дата обращения: 14.10.2025).

Регулирует дистанционный способ продажи товаров через интернет, обосновывая функционал корзины и подтверждения заказа.

4. Российская Федерация. Законы. О персональных данных [Электронный ресурс] : федер. закон от 27.07.2006 № 152-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_61801/. – (Дата обращения: 14.10.2025).

Регулирует сбор, хранение и обработку данных клиентов (ФИО, телефон, email), хранящихся в таблице «Пользователи».

5. Российская Федерация. Законы. Об информации, информационных технологиях и о защите информации [Электронный ресурс] : федер. закон от 27.07.2006 № 149-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_61798/. – (Дата обращения: 14.10.2025).

Устанавливает требования к защите информации в информационных системах и обязанности владельца сайта.

6. ГОСТ Р 59382-2021. Информационные технологии. Методы и средства обеспечения безопасности. Основы управления идентичностью [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/403448112/>. – (Дата обращения: 14.10.2025).

Регламентирует процессы регистрации, хранения паролей и разделения ролей (таблица «Роли»: Админ/Клиент).

7. ГОСТ Р ИСО 9241-210-2016. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/63110/>. – (Дата обращения: 14.10.2025).

Обосновывает выбор удобной навигации, наличие поиска и фильтрации ПО для снижения дискомфорта пользователя.

8. ГОСТ Р 52872-2019. Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Требования доступности для инвалидов по зрению [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/71634/>. – (Дата обращения: 14.10.2025).

Обосновывает реализацию интерфейса с учетом контрастности и возможности использования экранных дикторов (версия для слабовидящих).

9. ГОСТ Р 7.0.97-2016. Система стандартов по информации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов [Электронный ресурс] : утв. Приказом Росстандарта от 08.12.2016 № 2004-ст (ред. от 18.07.2023) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_216461/. – (Дата обращения: 14.10.2025).

Требования к оформлению документов, что важно для генерации электронных чеков и бланков лицензий после покупки.

10. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/11254/>. – (Дата обращения: 14.10.2025).

Определяет состав требований к функционалу (F1-F8), описанных в техническом задании.

11. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/59449/>. – (Дата обращения: 14.10.2025).

Обосновывает этапы разработки: от анализа БД до тестирования (функциональное, кросс-браузерное).

12. Об утверждении Перечня типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения [Электронный ресурс] : приказ Минкультуры России от 20.12.2019 № 236 (ред. от 24.02.2023) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_345020/. – (Дата обращения: 14.10.2025).

Определяет, как долго в базе данных должны храниться записи о совершенных транзакциях и заказах (таблицы «Заказы», «Состав_заказа»).

13. Налоговый кодекс Российской Федерации (часть вторая) [Электронный ресурс] : федер. закон от 05.08.2000 № 117-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа:

https://www.consultant.ru/document/cons_doc_LAW_28165/. – (Дата обращения: 14.10.2025).

Регулирует освобождение от НДС передачи прав на ПО (внесенное в реестр РФ), что обосновывает логику расчета итоговой суммы в таблице «Заказы».

14. ГОСТ Р ИСО/МЭК 27001-2021. Информационные технологии. Методы обеспечения безопасности. Системы менеджмента информационной безопасности. Требования [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200181890/>. – (Дата обращения: 14.10.2025).

Обосновывает применение мер защиты базы данных от несанкционированного доступа (SQL-инъекций, утечек).

Основные процессы продажи и управления ПО, автоматизируемые системой:

- **Идентификация пользователей:** Разграничение прав доступа для ролей «Администратор» и «Клиент».
- **Ведение каталога товаров:** Хранение подробной информации о программных продуктах, включая описание, системные требования, цены и категории (ОС, Офисные, Антивирусы и др.).
- **Управление корзиной и заказами:** Возможность временного хранения выбранных позиций в корзине с последующим оформлением заказа и расчетом итоговой суммы.
- **Генерация и учет лицензий:** Автоматическое закрепление лицензионных ключей за позициями заказа, фиксация даты активации и расчет даты истечения лицензии в зависимости от выбранного срока.

Входные данные:

- Данные о программных продуктах (название, цена, описание).
- Персональные данные клиентов при регистрации. [4]

- Параметры заказов (выбранные товары, количество, сроки лицензии).

Выходные данные:

- Лицензионные ключи для приобретенного ПО. [1, 9, 12]
- История заказов пользователя с их текущим статусом. [3, 9, 12, 13]
- Отчеты о продажах и популярности категорий товаров. [1, 2, 3, 13]

Сформулируем возможные вопросы, ответы на которые можно будет получить с помощью запросов к базе данных:

1. Какие товары входят в определенную категорию?
2. Какие товары представлены в магазине?
3. Какова итоговая стоимость всех товаров в корзине конкретного пользователя?
4. Какой программный продукт самый дешевой?
5. Когда был активирован ключ и когда истекает его срок действия?
6. Какая категория товаров приносит наибольшую выручку?
7. Какие товары самые популярные?

Анализ описания предметной области позволяет определить следующие независимые информационные объекты (сущности): ПОЛЬЗОВАТЕЛИ, КАТАЛОГ_ПО, КОРЗИНА, ЗАКАЗЫ.

Анализ описания предметной области позволяет выделить следующий набор данных (атрибутов), которые должны храниться в проектируемой базе данных:

1. **Пользователь (Пользователи) [4]**
 - Фамилия (Фамилия)
 - Имя (Имя)
 - Отчество (Отчество)
 - Телефон (Телефон)
 - Дата рождения (Дата_рождения)
 - Логин (Логин)
 - Пароль (Пароль)

- Роль (Роль)
- 2. **Программный продукт (Товар) [1, 2, 3, 13]**
 - Название ПО (Название)
 - Описание и системные требования (Описание)
 - Стоимость (Цена)
 - Категория (Категории)
 - Изображение (Изображение)
- 3. **Заказ (Заказ) [3, 9, 12, 13]**
 - Дата совершения покупки (Дата_заказа)
 - Текущий статус оплаты/доставки (Статус)
 - Общая стоимость заказа (Итоговая_сумма)
 - Тип лицензии
 - Товар
 - Количество
 - Пользователь
- 4. **Корзина (Корзина) [3, 7]**
 - Пользователь (Пользователь)
 - Товар (Товар)
 - Количество товара (Количество)

1.2 Модель сущность-связь

Используя информационные объекты (сущности) спроектируем модель «сущность-связь», представленную на рисунке 1.

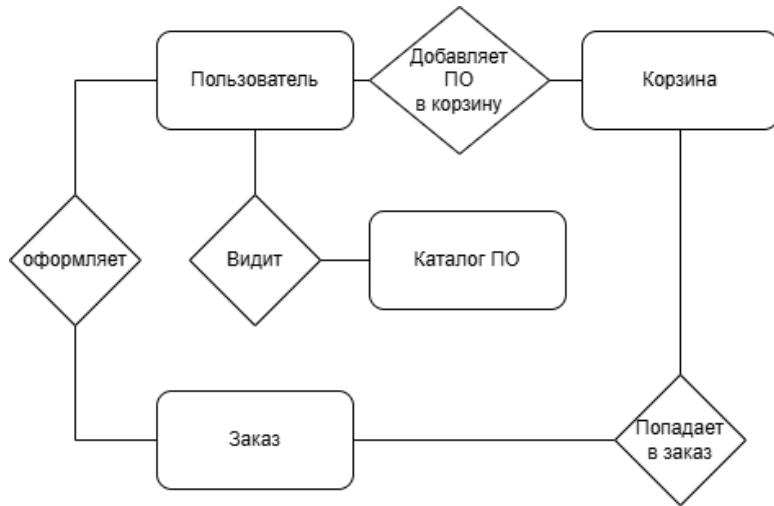


Рисунок 1 – Модель «Сущность-связь»

1.3 Дата-логическое (логическое) проектирование базы данных

Перейдем к созданию дата-логической модели базы данных. Для этого поставим в соответствие каждой сущности реляционную таблицу с соответствующими атрибутами и ключами.

Перед построением дата логической модели приведем нашу базу данных к третьей нормальной форме.

Нормализация к 1НФ не требуется, все атрибуты были атомизированы на этапе концептуального моделирования, согласно [1, 2, 4, 6, 12]

В процессе нормализации до 2НФ сформулированы идентификаторы к таблицам:

Выделены справочники: РОЛИ, КАТЕГОРИЯ, СОСТАВ_ЗАКАЗА, ЛИЦЕНЗИИ

В процессе формирования третьей нормальной формы были сформированы связи: в основной таблице ПОЛЬЗОВАТЕЛИ атрибут «Роль» был заменен на внешний ключ, ведущий на справочник РОЛИ. Аналогично, в таблице ТОВАРЫ атрибут «Категория» был заменен на внешний ключ, связывающий ее со справочником КАТЕГОРИИ.

Для обеспечения связи между пользователями и выбранными товарами в таблице КОРЗИНА атрибуты «Пользователь» и «Товар» были заменены на

внешние ключи, ведущие на таблицы ПОЛЬЗОВАТЕЛИ и ТОВАРЫ соответственно.

Структура данных о покупках была детализирована: в таблице СОСТАВ_ЗАКАЗА атрибуты «Заказ» и «Товар» были заменены на внешние ключи, что позволяет связывать позиции заказа с общей информацией о транзакции в таблице ЗАКАЗЫ и с характеристиками продукта в таблице ТОВАРЫ.

Наконец, для обеспечения полной прослеживаемости активации программного обеспечения в таблице ЛИЦЕНЗИИ атрибут «Позиция состава заказа» был заменен на внешний ключ, ведущий на конкретную запись в таблице СОСТАВ_ЗАКАЗА.

Дата-логическая модель представленную на рисунке 2.

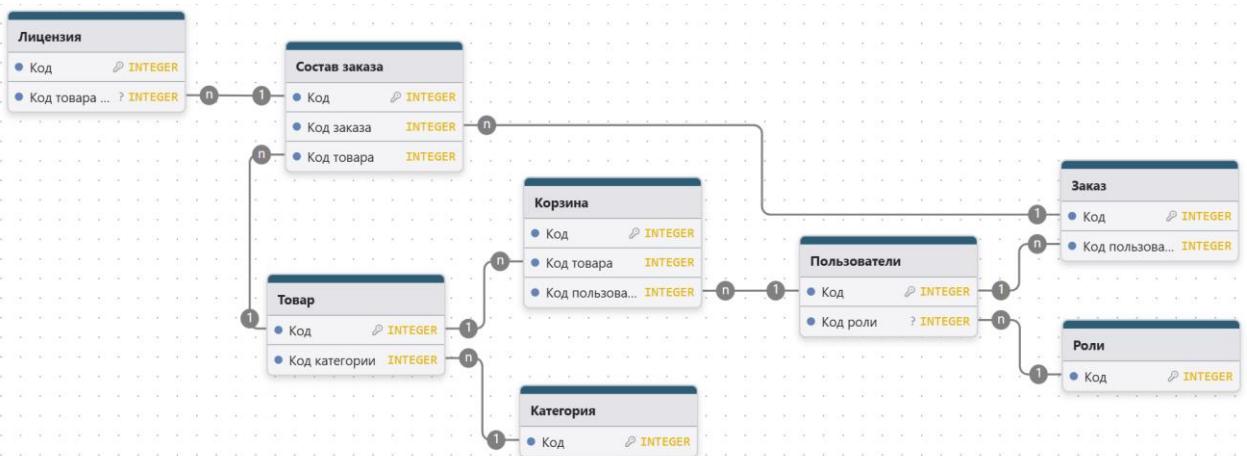


Рисунок 2 – Логическая структура базы данных

Рассмотрим описание логической модели базы данных в таблице 1.

Таблица 1 – Структура базы данных

Наименование таблицы	Описание таблицы
Пользователь	Содержит основную информацию о зарегистрированном пользователе (клиенте или администраторе).
Роль	Справочник, который определяет права доступа в системе (например, «Администратор», «Клиент»).
Товар	Содержит полное описание программного обеспечения: название, описание, цену, технические характеристики.
Категория	Справочник, который классифицирует программные продукты по типам или назначению (например, «Антивирусы», «Графические редакторы»).

Корзина	Хранит временный список товаров, отобранных пользователем для последующего оформления заказа.
Заказ	Содержит основную информацию о совершенной покупке: номер, дату, статус и итоговую сумму.
Состав_заказа	Детализирует заказ, связывая его с конкретными товарами, их количеством и ценой на момент покупки.
Лицензия	Хранит уникальные активационные ключи (лицензии), привязанные к конкретным позициям в составе заказа для их передачи покупателю.

1.4 Физическое моделирование базы данных

MySQL – это открытая реляционная система управления базами данных (СУБД), представляющая собой надежное и производительное решение для хранения структурированных данных. Как часть стека технологий LAMP (Linux, Apache, MySQL, PHP/Python/Perl), MySQL широко используется в веб-разработке благодаря своей стабильности, высокой производительности и простоте интеграции с различными языками программирования.

В контексте разрабатываемой системы "Программные продукты" MySQL была выбрана в качестве СУБД по следующим причинам:

- **Открытая лицензия (GPL)** позволяет использовать систему без дополнительных финансовых затрат
- **Полная поддержка SQL-стандарта** включает все необходимые функции: транзакции ACID, внешние ключи, хранимые процедуры, триггеры
- **Высокая производительность** при работе с типичными операциями: частые SELECT-запросы для анализа данных, INSERT/UPDATE операций
- **Широкая экосистема инструментов** включает MySQL Workbench для визуального проектирования БД, драйверы для интеграции с Python (pymysql)
- **Надежность и отказоустойчивость** обеспечиваются механизмами репликации, резервного копирования и восстановления данных

- **Масштабируемость** позволяет системе расти вместе с увеличением количества пользователей без значительных архитектурных изменений

Определим свойства полей для каждой таблицы. Тип данных в поле выбирается, исходя из характера данных, которые предполагается в нем хранить. Будем использовать типы данных, принятые в системе управления базами данных MySQL. Необходимо учесть, что поле, являющееся первичным ключом таблицы, не может содержать значений, повторяющихся в двух или более строках таблицы.

1) Таблицы

A) Таблица «Пользователи». [4]

Типы данных для полей таблицы «Пользователи» представлены в таблице 2.

Таблица 2 – Физическая модель таблицы «Пользователи» [4]

Имя поля в таблице	Тип данных	Описание
ID_Пользователя	INT PRIMARY KEY	Уникальный идентификатор пользователя
Фамилия	VARCHAR(50)	Фамилия пользователя
Имя	VARCHAR(50)	Имя пользователя
Отчество	VARCHAR(50)	Отчество пользователя
Телефон	VARCHAR(20)	Номер телефона
Дата рождения	DATE	Дата рождения
Логин	VARCHAR(50)	Логин, которым является почта
Пароль	VARCHAR(255)	Пароль профиля
ID_Роли	INT	Внешний ключ на таблицу «Роли»
Дата регистрации	TIMESTAMP	Дата и время создания профиля

Атрибут «ID_Пользователя» (первичный ключ таблицы) однозначно определяет конкретного клиента или администратора, а остальные атрибуты данного отношения индивидуальны относительно первичного ключа. Следовательно, таблица «Пользователи» содержит только одну функциональную зависимость от ключевого атрибута. Таким образом, таблица «Пользователи» удовлетворяет определению третьей нормальной формы (3NF), что исключает хранение избыточной информации и не требует дальнейшей декомпозиции.

Б) Таблица «Роли». [6, 14]

Типы данных для полей таблицы «Роли» представлены в таблице 3.

Таблица 3 – Физическая модель таблицы «Роли» [6, 14]

Имя поля в таблице	Тип данных	Описание
ID_Роли	INT PRIMARY KEY	Уникальный идентификатор роли
Название	VARCHAR(50)	Название

Атрибут «ID_Роли» (первичный ключ таблицы) однозначно определяет уровень доступа в системе. Таблица является нормализованным справочником. Единственный неключевой атрибут «Название» находится в полной функциональной зависимости от первичного ключа. В таблице отсутствуют транзитивные зависимости, что соответствует требованиям 3NF.

В) Таблица «Категории». [2, 3]

Типы данных для полей таблицы «Категории» представлены в таблице 4.

Таблица 4 – Физическая модель таблицы «Категории» [2, 3]

Имя поля в таблице	Тип данных	Описание
ID_Категории	INT PRIMARY KEY	Уникальный идентификатор категории
Название_категории	VARCHAR(100)	Полное название категории

Атрибут «ID_Категории» однозначно определяет группу товаров. Таблица содержит нормализованные данные о категориях ПО. Уникальность названий гарантирует отсутствие дублирования. Таблица удовлетворяет определению 3NF, так как не имеет частичных или транзитивных зависимостей.

Г) Таблица «Товары» [1, 2, 3, 13]

Типы данных для полей таблицы «Товары» представлены в таблице 5.

Таблица 5 – Физическая модель таблицы «Товары» [1, 2, 3, 13]

Имя поля в таблице	Тип данных	Описание
ID_Товара	INT	Уникальный идентификатор товара
Название	VARCHAR(255)	Название программного продукта
Описание	TEXT	Описание и характеристики программы
Цена	DECIMAL(10, 2)	Текущая цена лицензии ПО
ID_Категории	INT	Внешний ключ на таблицу «Категории»
Изображение	VARCHAR(255)	Путь к файлу изображения товара

Статус_активности	BOOLEAN	Если товар актуален и должен отображаться в каталоге, то его статус = 1, если его нужно удалить(отправить в архив) статус = 0.
-------------------	---------	--

Атрибут «ID_Товара» однозначно идентифицирует программный продукт. Все неключевые атрибуты (цена, описание) зависят только от первичного ключа. Внешний ключ ID_Категории обеспечивает связь со справочником. Таблица соответствует 3NF, обеспечивая отсутствие аномалий при обновлении информации о ПО.

Д) Таблица «Корзина» [3, 7]

Типы данных для полей таблицы «Корзина» представлены в таблице 6.

Таблица 6 – Физическая модель таблицы «Корзина»

Имя поля в таблице	Тип данных	Описание
ID_Корзины	INT PRIMARY KEY	Уникальный идентификатор записи в корзине
ID_Пользователя	INT	Внешний ключ на таблицу «Пользователи»
ID_Товара	INT	Внешний ключ на таблицу «Товары»
Количество	INT	Количество единиц выбранного товара

В таблице «Корзина» атрибуты «ID_Пользователя» и «ID_Товара» являются внешними ключами. Структура таблицы позволяет хранить временные данные о выборе клиента до момента оформления заказа. Все атрибуты функционально зависят от первичного ключа записи. Таблица находится в 3NF.

Е) Таблица «Заказы» [3, 9, 12, 13]

Типы данных для полей таблицы «Заказы» представлены в таблице 7.

Таблица 7 – Физическая модель таблицы «Заказы» [3, 9, 12, 13]

Имя поля в таблице	Тип данных	Описание
ID_Заказа	INT PRIMARY KEY	Уникальный идентификатор
ID_Пользователя	INT	Внешний ключ на таблицу «Пользователи»
Дата заказа	DATETIME	Дата и время совершения покупки
Статус	VARCHAR(50)	Статус (Оплачен, Обработка)
Итоговая_сумма	DECIMAL(10, 2)	Общая стоимость всего заказа

Атрибут «ID_Заказа» однозначно определяет операцию покупки. Таблица является основной операционной таблицей системы. Внешний ключ связывает покупку с конкретным пользователем. Отсутствие транзитивных зависимостей подтверждает соответствие таблицы требованиям 3NF.

Ж) Таблица «Состав_заказа» [1, 12, 13]

Типы данных для полей таблицы «Состав_заказа» представлены в таблице 8.

Таблица 8 – Физическая модель таблицы «Состав_заказа» [1, 12, 13]

Имя поля в таблице	Тип данных	Описание
ID_Позиции	INT PRIMARY KEY	Уникальный идентификатор периода
ID_Заказа	INT	Внешний ключ на таблицу «Заказы»
ID_Товара	INT	Внешний ключ на таблицу «Товары»
Цена_продажи	DECIMAL(10, 2)	Цена продукта на момент покупки
Срок лицензии_дни	INT	Срок лицензии (в днях)
Количество	INT	Количество приобретенных лицензий

Данная таблица необходима для разрешения связи «многие-ко-многим» между заказами и товарами. Атрибут «Цена_продажи» фиксирует стоимость на момент транзакции, что исключает зависимость итогов заказа от будущих изменений цен в каталоге. Таблица удовлетворяет 3NF.

Ж) Таблица «Лицензии» [1, 9, 12]

Типы данных для полей таблицы «Лицензии» представлены в таблице 9.

Таблица 9 – Физическая модель таблицы «Лицензии» [1, 9, 12]

Имя поля в таблице	Тип данных	Описание
ID_Лицензии	INT PRIMARY KEY	Уникальный идентификатор ключа
ID_Позиции_заказа	INT	Внешний ключ на таблицу «Состав_заказа»
Лицензионный_ключ	VARCHAR(255)	Уникальный лицензионный код
Дата активации	VARCHAR(255)	Дата начала действия ключа
Дата истечения	INT	Дата окончания срока лицензии

Атрибут «ID_Лицензии» однозначно идентифицирует цифровую лицензию. Связь через ID_Позиции позволяет определить, в рамках какого заказа был выдан данный ключ. Все неключевые атрибуты зависят только от первичного ключа, что соответствует 3NF и обеспечивает корректный учет сроков действия ПО.

Связи между таблицами

Имена таблиц расположены в порядке указания направления связи. Например, Роли 1 – Пользователи Много, где тип связи «один ко многим», обозначается как 1..M.

A) Зависимость «Роли – Пользователи». Связь можно описать выражением:

- «Каждый идентификатор пользователя должен иметь идентификатор роли и при том только один».
- «Каждый идентификатор роли может быть связан с различными пользователями».

Таким образом, связь таблиц «Роли» и «Пользователи» имеет мощность **один – ко – многим** и является **обязательной**.

B) Зависимость «Категории – Товары». Связь можно описать выражением:

- «Каждый идентификатор товара должен иметь идентификатор категории и при том только один».
- «Каждый идентификатор категории может быть связан с различными товарами».

Таким образом, связь таблиц «Категории» и «Товары» имеет мощность **один – ко – многим** и является **обязательной**.

C) Зависимость «Пользователи – Корзина». Связь можно описать выражением:

- «Каждый идентификатор записи в корзине должен иметь идентификатор пользователя и при том только один».
- «Каждый идентификатор пользователя может быть связан с различными записями в корзине (несколькими товарами)».

Таким образом, связь таблиц «Пользователи» и «Корзина» имеет мощность **один – ко – многим** и является **обязательной** для записи в корзине.

D) Зависимость «Товары – Корзина». Связь можно описать выражением:

- «Каждый идентификатор записи в корзине должен иметь идентификатор товара и при том только один».
- «Каждый идентификатор товара может быть связан с записями в корзинах различных пользователей».

Таким образом, связь таблиц «Товары» и «Корзина» имеет мощность **один – ко – многим** и является **обязательной** для формирования корзины.

Д) Зависимость «Пользователи – Заказы». Связь можно описать выражением:

- «Каждый идентификатор заказа должен иметь идентификатор пользователя и при том только один».
- «Каждый идентификатор пользователя может быть связан с различными заказами (история покупок)».

Таким образом, связь таблиц «Пользователи» и «Заказы» имеет мощность **один – ко – многим** и является **обязательной**.

Е) Зависимость «Заказы – Состав_заказа». Связь можно описать выражением:

- «Каждый идентификатор позиции в составе заказа должен иметь идентификатор заказа и при том только один».
- «Каждый идентификатор заказа может быть связан с различными записями в составе заказа (несколько программ в одном чеке)».

Таким образом, связь таблиц «Заказы» и «Состав_заказа» имеет мощность **один – ко – многим** и является **обязательной**.

Ж) Зависимость «Товары – Состав_заказа». Связь можно описать выражением:

- «Каждый идентификатор позиции в составе заказа должен иметь идентификатор товара и при том только один».
- «Каждый идентификатор товара может быть связан с различными записями о продажах (составами разных заказов)».

Таким образом, связь таблиц «Товары» и «Состав_заказа» имеет мощность **один – ко – многим** и является **обязательной**.

3) Зависимость «Состав_заказа – Лицензии». Связь можно описать выражением:

- «Каждый идентификатор лицензии должен иметь идентификатор позиции состава заказа и при том только один».
- «Каждый идентификатор позиции состава заказа может быть связан с различными лицензиями (если куплено несколько штук одного ПО)».

Таким образом, связь таблиц «Состав_заказа» и «Лицензии» имеет мощность **один – ко – многим** и является **обязательной**.

Физическое проектирование — создание схемы базы данных для конкретной СУБД с учетом ее специфики. Результатом физического проектирования логической схемы может быть скрипт на языке SQL.

Перейдем к физическому проектированию базы данных. Типовой пример реализации создания таблиц и отображения содержимого таблицы в базе данных представлен на рисунке 3

```
1 • CREATE TABLE Пользователи (
2     ID_Пользователя INT PRIMARY KEY AUTO_INCREMENT,
3     Фамилия VARCHAR(50) NOT NULL,
4     Имя VARCHAR(50),
5     Отчество VARCHAR(50),
6     Телефон VARCHAR(20) UNIQUE,
7     Дата_рождения DATE,
8     Логин VARCHAR(50) NOT NULL UNIQUE,
9     Пароль VARCHAR(255) NOT NULL,
10    ID_Роли INT,
11    Дата_регистрации TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
12    FOREIGN KEY (ID_Роли) REFERENCES Роли(ID_Роли)
13 );
```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window displaying the SQL script for creating the 'Пользователи' table. Below it is a 'Result Grid' window showing the data inserted into the table. The table has columns: ID_Пользователя, Фамилия, Имя, Отчество, Телефон, Дата_рождения, Логин, Пароль, ID_Роли, and Дата_регистрации. The data grid contains five rows of data, with the fifth row being a header row indicated by an asterisk (*).

*	ID_Пользователя	Фамилия	Имя	Отчество	Телефон	Дата_рождения	Логин	Пароль	ID_Роли	Дата_регистрации
1		Иванов	Иван	Иванович	NULL	NULL	admin@key.ru	admin123	1	2026-01-04 20:29:55
2		Петров	Алексей	Сергеевич	NULL	NULL	petrov_as@key.ru	password55	2	2026-01-04 20:29:55
3		Сидорова	Мария	Павловна	NULL	NULL	masha_99@key.ru	123456	2	2026-01-04 20:29:55
4		Клиент	Роман	NULL	NULL	NULL	roman@key.ru	admin123	2	2026-01-05 19:03:16

Рисунок 3 – Таблица «Пользователи»

Реализация схемы данных приведена на рисунке 4.

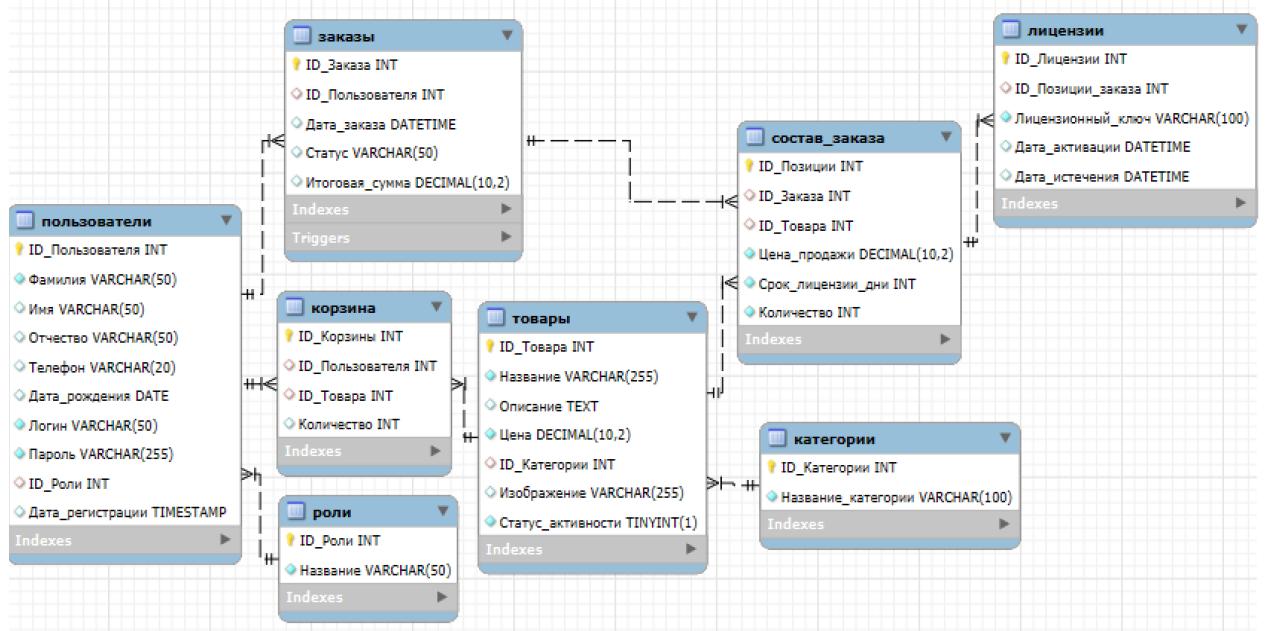


Рисунок 4 – Схема данных

2 Функциональные возможности

Вопросы, обозначенные при анализе предметной области, преобразуются в функциональные элементы (функции базы данных), являющиеся запросами, процедурами, триггерами. Приведём основные типовые запросы, полный перечень реализован в коде приложения.

2.1 Запросы на редактирование

Запросы SQL к базе данных используют синтаксические особенности языка программирования Python (библиотека PyMySQL), что позволяет использовать динамические обращения к полям базы данных в зависимости от собираемых данных.

2.1.1 Запрос на добавление записи в таблицу

Рассмотрим запрос на добавление нового программного продукта через панель администратора. Запрос вставляет данные в таблицу Товары, включая название, цену, описание и путь к изображению. Запрос представлен на рисунке 5.

```
sql = "INSERT INTO Товары (Название, Цена, ID_Категории, Описание, Изображение) VALUES (%s, %s, %s, %s, %s)"
```

Рисунок 5 – Запрос на добавление нового программного продукта

2.1.2 Запрос на изменение данных таблицы

Типовой пример — обновление профиля пользователя (фамилия, имя, телефон). Перед выполнением UPDATE система проверяет, не занят ли новый номер телефона другим пользователем, чтобы не нарушить уникальность записей. Запрос представлен на рисунке 6.

```
sql = """
UPDATE Пользователи
SET Фамилия = %s, Имя = %s, Отчество = %s, Телефон = %s, Дата_рождения = %s
WHERE ID_Пользователя = %s
"""
```

Рисунок 6 – Запрос на обновление профиля

По схожему принципу с предварительными проверками построен запрос обновления пароля.

2.1.3 Запрос на удаление данных из таблицы

В системе реализовано удаление записей из корзины. Запрос выполняется при нажатии кнопки «Удалить» в интерфейсе и очищает таблицу Корзина для конкретного пользователя и товара. Запрос представлен на рисунке 7.

```
cursor.execute("DELETE FROM Корзина WHERE ID_Пользователя = %s AND ID_Товара = %s", (session['user_id'], product_id))
```

Рисунок 7 – Запрос на удаление продукта из корзины

2.2 Запросы на выборку данных Запрос на фильтрацию данных из таблицы

Рассмотрим пример сложного запроса на выборку данных, который имеет динамическую структуру и используется для построения детализированного каталога товаров. Основной запрос главной страницы осуществляет выборку товаров с объединением (JOIN) таблицы Категории. Это позволяет отображать текстовое название категории. Запрос поддерживает динамическую фильтрацию по категории, поисковому слову и сортировку по цене. Запрос представлен на рисунке 8.

```

# Получаем все категории для выпадающего списка
cursor.execute("SELECT * FROM Категории")
categories = cursor.fetchall()

# Строим запрос
sql = "SELECT t.*, k.Название_категории FROM Товары t JOIN Категории k ON t.ID_Категории = k.ID_Категории WHERE 1=1"
params = []

if category_id and category_id != 'all':
    sql += " AND t.ID_Категории = %s"
    params.append(category_id)
if search_query:
    sql += " AND t.Название LIKE %s"
    params.append(f"%{search_query}%")

# Сортировка
order_clauses = []
if sort_price == 'asc': order_clauses.append("t.Цена ASC")
elif sort_price == 'desc': order_clauses.append("t.Цена DESC")

if sort_date == 'new': order_clauses.append("t.ID_Товара DESC")
elif sort_date == 'old': order_clauses.append("t.ID_Товара ASC")

if order_clauses:
    sql += " ORDER BY " + ", ".join(order_clauses)

```

Рисунок 8 – Запрос на фильтрацию данных

В отличие от статических запросов, этот запрос сначала формирует базовую, неизменяемую часть запроса SELECT...FROM...JOIN...WHERE 1=1. Конструкция WHERE 1=1 используется для удобства последующего добавления условий. Затем, в зависимости от действий пользователя в интерфейсе (выбор фильтра по категории, ввод поискового запроса, установка параметров сортировки), к строке SQL-запроса динамически добавляются соответствующие условия (AND) и операторы сортировки (ORDER BY). Параллельно в отдельный список добавляются значения параметров, которые безопасно подставляются в запрос на этапе его выполнения. Такой подход, известный как параметризованные запросы, позволяет избежать угроз SQL-инъекций, а также минимизировать передачу в СУБД избыточных данных, что обеспечивает высокую скорость отклика при фильтрации и сортировке большого объема товарной информации.

2.2.2 Запрос с использованием агрегатных функций

Для расчета выручки по категориям программного обеспечения используется агрегатная функция SUM() в сочетании с группировкой данных. База данных выполняет объединение (JOIN) нескольких таблиц:

Состав_заказа, Товары и Категории. Для каждой категории система суммирует общую выручку, которая рассчитывается как произведение цены продажи (Цена_продажи) на количество единиц (Количество) для всех позиций заказа, относящихся к товарам данной категории. Запрос представлен на рисунке 9.

```
points_sql = """
SELECT SUM(rc.points * sa.quantity) as category_total
FROM Student_Achievement sa
JOIN Rating_Criteria rc ON sa.criteria_id = rc.id
JOIN Academic_Period ap ON sa.period_id = ap.id
WHERE sa.student_id = %s AND rc.category_id = %s
AND ap.end_date >= CURDATE()
"""
```

Рисунок 9 – Запрос на формирование анализа выручки по категориям

2.2.3 Запрос с объединением нескольких таблиц (JOIN)

Для формирования истории заказов в личном кабинете используется LEFT JOIN, который связывает таблицы Заказы, Состав_заказа, Товары и Лицензии. Это позволяет в одном списке вывести дату заказа, название программы и выданный лицензионный ключ. Запрос представлен на рисунке 10.

```
# Получаем заказы, товары и их ключи одним запросом
cursor.execute("""
SELECT
    z.ID_Заказа, z.Дата_заказа, z.Статус, z.Итоговая_сумма,
    sz.ID_Позиции, sz.Количество, sz.Цена_продажи,
    t.Название,
    l.Лицензионный_ключ
FROM Заказы z
JOIN Состав_заказа sz ON z.ID_Заказа = sz.ID_Заказа
JOIN Товары t ON sz.ID_Товара = t.ID_Товара
LEFT JOIN Лицензии l ON sz.ID_Позиции = l.ID_Позиции_заказа
WHERE z.ID_Пользователя = %s
ORDER BY z.Дата_заказа DESC
""", (session['user_id'],))
```

Рисунок 10 – Запрос на получение данных заказа и ключей

2.2.4 Запрос на проверку уникальности данных

Перед выполнением регистрации нового пользователя или сохранением изменений в личном кабинете, система инициирует проверочную выборку для поиска дубликатов в полях, имеющих ограничение уникальности (UNIQUE). К таким полям в базе данных относятся логин и номер телефона. Запрос на выборку использует оператор OR для одновременной проверки нескольких параметров. В случае редактирования профиля в запрос добавляется условие исключения, текущего ID_Пользователя, чтобы система не считала ошибкой совпадение данных с текущими данными этого же пользователя. Если запрос возвращает результат, операция прерывается с выводом ошибки, что гарантирует целостность данных. Запрос представлен на рисунке 11.

```
# Проверяем, нет ли уже такого логина
cursor.execute("SELECT * FROM Пользователи WHERE Логин = %s", (email,))
if cursor.fetchone():
    flash('Пользователь с такой почтой уже существует', 'error')
```

Рисунок 11 – Запрос на проверку уникальности данных пользователя

2.3 Триггеры

Триггер «Расчет_итоговой_суммы» срабатывает перед добавлением новой записи в таблицу заказов (Заказы). Его задача — автоматически рассчитать итоговую сумму заказа, если она не была указана явно, и обеспечить валидацию данных. Триггер проверяет, чтобы итоговая сумма не была отрицательной. Если при вставке указывается отрицательное значение, триггер предотвращает операцию и возвращает пользовательскую ошибку с соответствующим сообщением. Это позволяет поддерживать финансовую корректность данных в системе учета продаж. На рисунке 12 представлена блок-схема триггера, триггер представлен на рисунке 13.

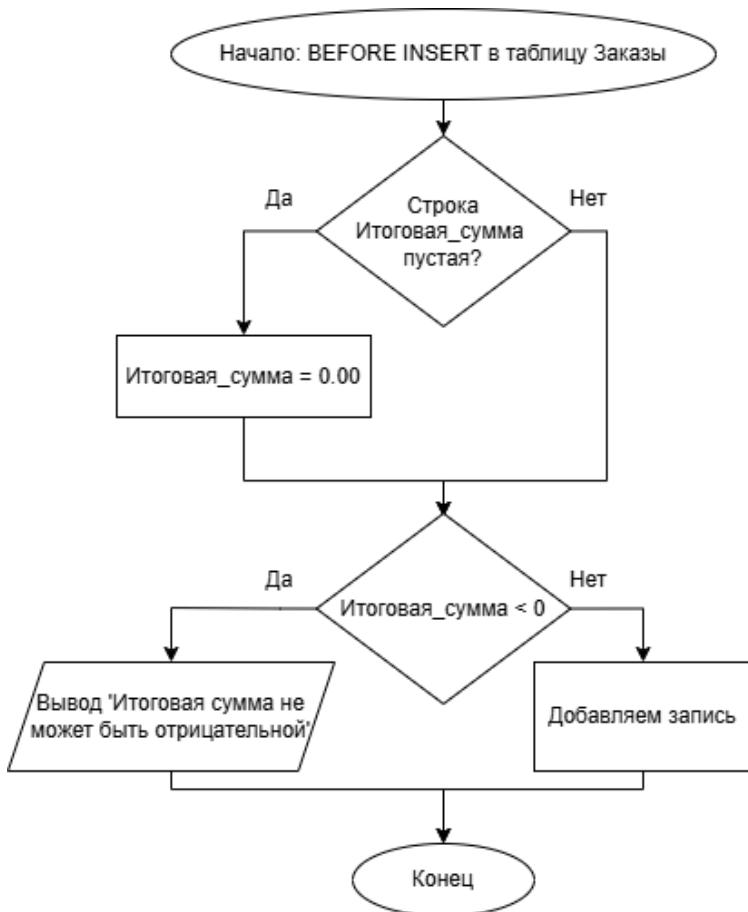


Рисунок 12 – Блок-схема триггера

```

DELIMITER //

CREATE TRIGGER Расчет_итоговой_суммы
BEFORE INSERT ON Заказы
FOR EACH ROW
BEGIN
    DECLARE v_сумма DECIMAL(10,2);

    -- Если сумма не указана
    IF NEW.Итоговая_сумма IS NULL THEN
        SET NEW.Итоговая_сумма = 0.00;
    END IF;

    -- Проверяем, что сумма не отрицательная
    IF NEW.Итоговая_сумма < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Итоговая сумма не может быть отрицательной';
    END IF;
END //

```

DELIMITER ;

Рисунок 13 – Триггер проверки итоговой суммы заказа

2.4 Процедуры

Рассмотрим процедуру, которая принимает идентификатор пользователя и срок действия лицензии, осуществляет полный процесс оформления заказа. Процедура выполняет следующие действия: рассчитывает общую стоимость товаров в корзине пользователя, создает запись в таблице заказов, переносит все позиции из корзины в детализированный состав заказа с фиксацией цен, генерирует записи о лицензионных ключах и полностью очищает корзину пользователя. Это позволяет автоматизировать процесс покупки и обеспечить атомарность транзакции. На рисунке 14 представлена блок-схема процедуры, процедура представлена на рисунке 15.

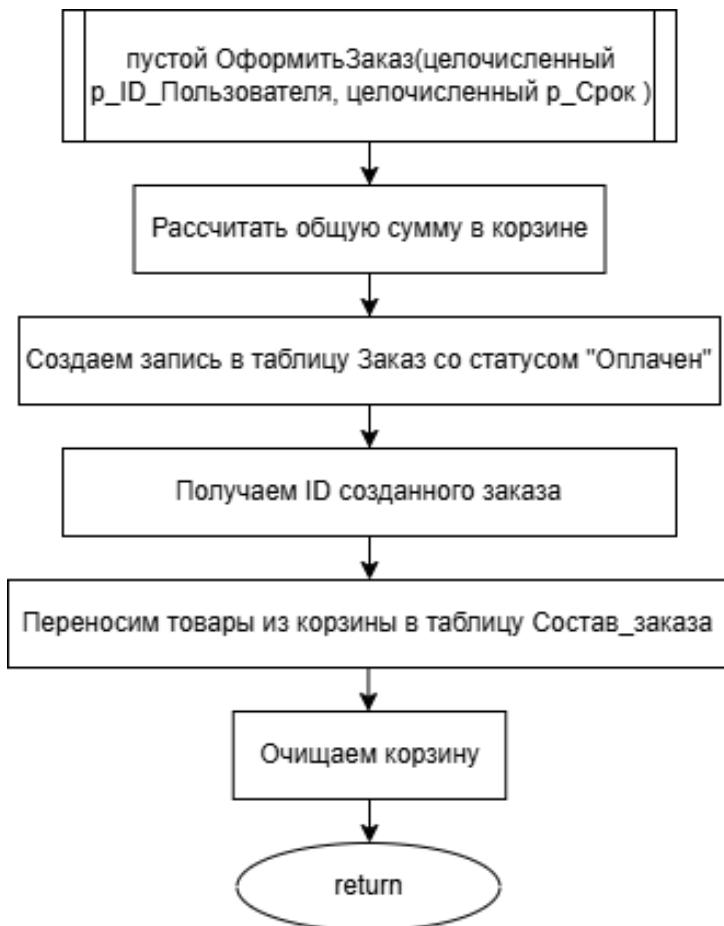


Рисунок 14 – Блок-схема процедуры

```

DELIMITER //
CREATE PROCEDURE ОформитьЗаказ(IN p_ID_Пользователя INT, IN p_Срок INT)
BEGIN
    DECLARE v_ID_Заказа INT;
    DECLARE v_Total DECIMAL(10,2);

    -- Считаем общую сумму в корзине
    SELECT SUM(T.Цена * К.Количество) INTO v_Total
    FROM Корзина K JOIN Товары T ON K.ID_Товара = T.ID_Товара
    WHERE K.ID_Пользователя = p_ID_Пользователя;

    -- Создаем запись в Заказах
    INSERT INTO Заказы (ID_Пользователя, Итоговая_сумма, Статус)
    VALUES (p_ID_Пользователя, v_Total, 'Оплачено');

    SET v_ID_Заказа = LAST_INSERT_ID();

    -- Переносим товары из корзины в Состав_заказа
    INSERT INTO Состав_заказа (ID_Заказа, ID_Товара, Цена_продажи, Срок_лицензии_дни, Количество)
    SELECT v_ID_Заказа, ID_Товара,
           (SELECT Цена FROM Товары WHERE ID_Товара = Корзина.ID_Товара),
           p_Срок, Количество
    FROM Корзина WHERE ID_Пользователя = p_ID_Пользователя;

    -- Очищаем корзину
    DELETE FROM Корзина WHERE ID_Пользователя = p_ID_Пользователя;
END //
DELIMITER ;

```

Рисунок 15 – Процедура формирования заказа

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Гражданский кодекс Российской Федерации (часть четвертая) [Электронный ресурс] : федер. закон от 18.12.2006 № 230-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_64629/. – (Дата обращения: 14.10.2025).
2. Российская Федерация. Законы. О защите прав потребителей [Электронный ресурс] : закон от 07.02.1992 № 2300-1 (ред. от 25.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_305/. – (Дата обращения: 14.10.2025).
3. Об утверждении Правил продажи товаров по договору розничной купли-продажи, перечня товаров длительного пользования, на которые не распространяется требование покупателя о безвозмездном предоставлении ему на период ремонта или замены аналогичного товара, и перечня непродовольственных товаров надлежащего качества, не подлежащих обмену, а также о внесении изменений в некоторые акты Правительства Российской Федерации [Электронный ресурс] : постановление Правительства Рос. Федерации от 31.12.2020 № 2463 (ред. от 01.04.2024) // Гарант. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_373622/. – (Дата обращения: 14.10.2025).
4. Российская Федерация. Законы. О персональных данных [Электронный ресурс] : федер. закон от 27.07.2006 № 152-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_61801/. – (Дата обращения: 14.10.2025).
5. Российская Федерация. Законы. Об информации, информационных технологиях и о защите информации [Электронный

ресурс] : федер. закон от 27.07.2006 № 149-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_61798/. – (Дата обращения: 14.10.2025).

6. ГОСТ Р 59382-2021. Информационные технологии. Методы и средства обеспечения безопасности. Основы управления идентичностью [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/403448112/>. – (Дата обращения: 14.10.2025).

7. ГОСТ Р ИСО 9241-210-2016. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/63110/>. – (Дата обращения: 14.10.2025).

8. ГОСТ Р 52872-2019. Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Требования доступности для инвалидов по зрению [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/71634/>. – (Дата обращения: 14.10.2025).

9. ГОСТ Р 7.0.97-2016. Система стандартов по информации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов [Электронный ресурс] : утв. Приказом Росстандарта от 08.12.2016 № 2004-ст (ред. от 18.07.2023) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_216461/. – (Дата обращения: 14.10.2025).

10. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/11254/>. – (Дата обращения: 14.10.2025).

11. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/59449/>. – (Дата обращения: 14.10.2025).

12. Об утверждении Перечня типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения [Электронный ресурс] : приказ Минкультуры России от 20.12.2019 № 236 (ред. от 24.02.2023) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_345020/. – (Дата обращения: 14.10.2025).

13. Налоговый кодекс Российской Федерации (часть вторая) [Электронный ресурс] : федер. закон от 05.08.2000 № 117-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_28165/. – (Дата обращения: 14.10.2025).

14. ГОСТ Р ИСО/МЭК 27001-2021. Информационные технологии. Методы обеспечения безопасности. Системы менеджмента информационной безопасности. Требования [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200181890/>. – (Дата обращения: 14.10.2025).