



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ (ИУ)

КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОЙ РАБОТЕ*

*НА ТЕМУ:*

*Разработка базы данных цифрового магазина  
музыкальных инструментов*

Студент группы ИУ7-62Б

---

Глотов И. А.

Руководитель курсовой работы

---

Кострицкий А. С.

2023 г.

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

---

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7

Рудаков И. В.

«3» марта 2023 г.

## **З А Д А Н И Е на выполнение курсовой работы**

по дисциплине

**Базы данных**

Студент группы **ИУ7-62Б**

**Глотов Илья Анатольевич**

Тема курсовой работы

**Разработка базы данных цифрового магазина музыкальных инструментов**

График выполнения работы: 25% к 4 нед., 50% к 8 нед., 75% к 13 нед., 100% к 15 нед.

### **Задание**

*Провести анализ предметной области цифровых магазинов музыкальных инструментов. Сформулировать требования и ограничения к разрабатываемой базе данных и приложению для цифрового магазина музыкальных инструментов. Провести анализ существующих решений. Сформулировать описание пользователей проектируемого приложения для доступа к базе данных. Спроектировать архитектуру базы данных и ограничения целостности. Спроектировать ролевую модель на уровне базы данных. Выбрать средства реализации. Реализовать спроектированную БД и необходимый интерфейс для взаимодействия с ней. Исследовать характеристики разработанного программного обеспечения.*

### **Оформление курсовой работы:**

Расчетно-пояснительная записка на 25-40 листах формата А4. Презентация на 12-18 слайдах.

Дата выдачи задания «3» марта 2023 г.

**Руководитель курсовой работы**

**Кострицкий А. С.**

**Студент**

**Глотов И. А.**

# **РЕФЕРАТ**

Расчетно-пояснительная записка 35 с., 12 рис., 10 табл., 14 источн., 1 прил.

Ключевые слова: база данных, магазин музыкальных инструментов, PostgreSQL, Golang.

Цель работы — проектирование и разработка базы данных для цифрового магазина музыкальных инструментов.

# СОДЕРЖАНИЕ

<b>РЕФЕРАТ</b>	<b>3</b>
<b>ВВЕДЕНИЕ</b>	<b>6</b>
<b>1 Аналитический раздел</b>	<b>8</b>
1.1 Требования и ограничения к разрабатываемой базе данных и приложению . . . . .	8
1.2 Анализ существующих решений . . . . .	9
1.3 Пользователи приложения для доступа к базе данных . . . . .	9
1.4 Формализация данных . . . . .	11
1.5 Анализ моделей данных . . . . .	11
1.5.1 Иерархическая модель данных . . . . .	12
1.5.2 Сетевая модель данных . . . . .	12
1.5.3 Реляционная модель данных . . . . .	13
1.6 Анализ систем управления базами данных . . . . .	13
<b>2 Конструкторский раздел</b>	<b>15</b>
2.1 Формализация сущностей системы . . . . .	15
2.2 Проектирование триггера . . . . .	19
2.3 Ролевая модель . . . . .	20
<b>3 Технологический раздел</b>	<b>21</b>
3.1 Разработка сущностей системы . . . . .	21
3.2 Разработка триггера . . . . .	21
3.3 Разработка ролевой модели . . . . .	21
3.4 Тестирование . . . . .	21
3.5 Интерфейс приложения . . . . .	22
<b>4 Исследовательский раздел</b>	<b>26</b>
4.1 Технические характеристики устройства . . . . .	26
4.2 Измерение времени выполнения запросов доступа к метаданным	26
4.3 Выводы из исследовательского раздела . . . . .	27
<b>ЗАКЛЮЧЕНИЕ</b>	<b>28</b>

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 30

**ПРИЛОЖЕНИЕ А** 31

# ВВЕДЕНИЕ

Наиболее перспективным направлением инноваций в конкурентных условиях является создание интернет-услуг, поскольку сегодня всемирная паутина предлагает множество современных инструментов, применимых в розничном секторе. Сегодня многие крупные и средние реальные магазины имеют свои собственные сайты, которые являются для них одним из основных каналов распространения информации о товаре, цене и других характеристиках [1].

Целью данной работы является проектирование и разработка базы данных для цифрового магазина музыкальных инструментов.

Для достижения поставленной цели требуется решить следующие задачи.

- 1) Сформулировать требования и ограничения к разрабатываемой базе данных и приложению для цифрового магазина музыкальных инструментов.
- 2) Провести анализ существующих решений.
- 3) Сформулировать описание пользователей проектируемого приложения для доступа к базе данных.
- 4) Формализовать данные, хранящиеся в базе данных.
- 5) Провести анализ моделей данных и выбрать подходящую модель.
- 6) Провести анализ систем управления базами данных и выбрать подходящую систему.
- 7) Спроектировать сущности базы данных и ограничения целостности цифрового магазина музыкальных инструментов.
- 8) Спроектировать триггер такой, что если удаляется бренд музыкальных инструментов, то удаляются все товары, связанные с этим брендом.
- 9) Спроектировать ролевую модель на уровне базы данных цифрового магазина музыкальных инструментов.

- 10) Разработать сущности базы данных цифрового магазина музыкальных инструментов и спроектированные ограничения целостности базы данных.
- 11) Разработать спроектированный триггер такой, что если удаляется бренд музыкальных инструментов, то удаляются все товары, связанные с этим брендом.
- 12) Разработать спроектированную роевую модель на уровне базы данных цифрового магазина музыкальных инструментов.
- 13) Описать методы тестирования разработанного функционала и разработать тесты для проверки корректности работы приложения.
- 14) Описать интерфейс доступа к базе данных.
- 15) Провести сравнительный анализ методов доступа к метаданным: с помощью системных таблиц, системных хранимых процедур и представлений информационной схемы.

# **1 Аналитический раздел**

## **1.1 Требования и ограничения к разрабатываемой базе данных и приложению**

Согласно техническому заданию, к базе данных предъявляются следующие требования.

- 1) Наличие ролевой модели на уровне базы данных (три роли).
- 2) Шесть сущностей в базе данных:
  - пользователь;
  - музыкальный инструмент;
  - бренд;
  - корзина;
  - заказ;
  - элемент заказа.
- 3) Наличие триггера такого, что если удаляется бренд музыкальных инструментов, то удаляются все товары, связанные с этим брендом.

Согласно техническому заданию, к приложению предъявляются следующие требования.

- 1) Возможность просмотра ассортимента товаров.
- 2) Возможность добавления товаров в корзину и удаления товаров из нее.  
Пользователь имеет возможность просматривать свою корзину, в ней производится сравнение товаров в виде сетки.
- 3) Возможность оформления заказа.
- 4) Возможность модификации информации в базе данных (добавление, удаление, обновление информации о музыкальных инструментах).

## 1.2 Анализ существующих решений

Сравнение существующих решений будет проводится по следующим критериям.

- 1) Возможность добавления товаров в корзину, в которой предусмотрено сравнение товаров в виде сетки.
- 2) Возможность просмотра ассортимента товаров.
- 3) Возможность оформления заказа.
- 4) Доступность в РФ.

В таблице 1.1 приведено сравнение существующих решений по заданным выше критериям.

Таблица 1.1 – Сравнение существующих решений

Решение	1	2	3	4
Музторг	-	+	+	+
Глинки.ру	-	+	+	+
Music Store	+	+	+	-
Musician's Friend	-	+	+	-
Разрабатываемое решение	+	+	+	+

## 1.3 Пользователи приложения для доступа к базе данных

Для работы с приложением для доступа к базе данных необходимым этапом является прохождение персонализации. Пользователь может работать в системе под одной из следующих ролей.

- 1) Незарегистрированный пользователь – пользователь, обладающий возможностями просмотра ассортимента товаров, регистрации и авторизации.
- 2) Зарегистрированный пользователь – пользователь, обладающий возможностями просмотра ассортимента товаров, добавления товаров в корзину, просмотра корзины, удаление товаров из корзины и оформления заказа.

- 3) Администратор – пользователь, обладающий всеми возможностями зарегистрированного пользователя, а также возможностью модификации информации в базе данных.

На рисунке 1.1 приведена диаграмма использования приложения.

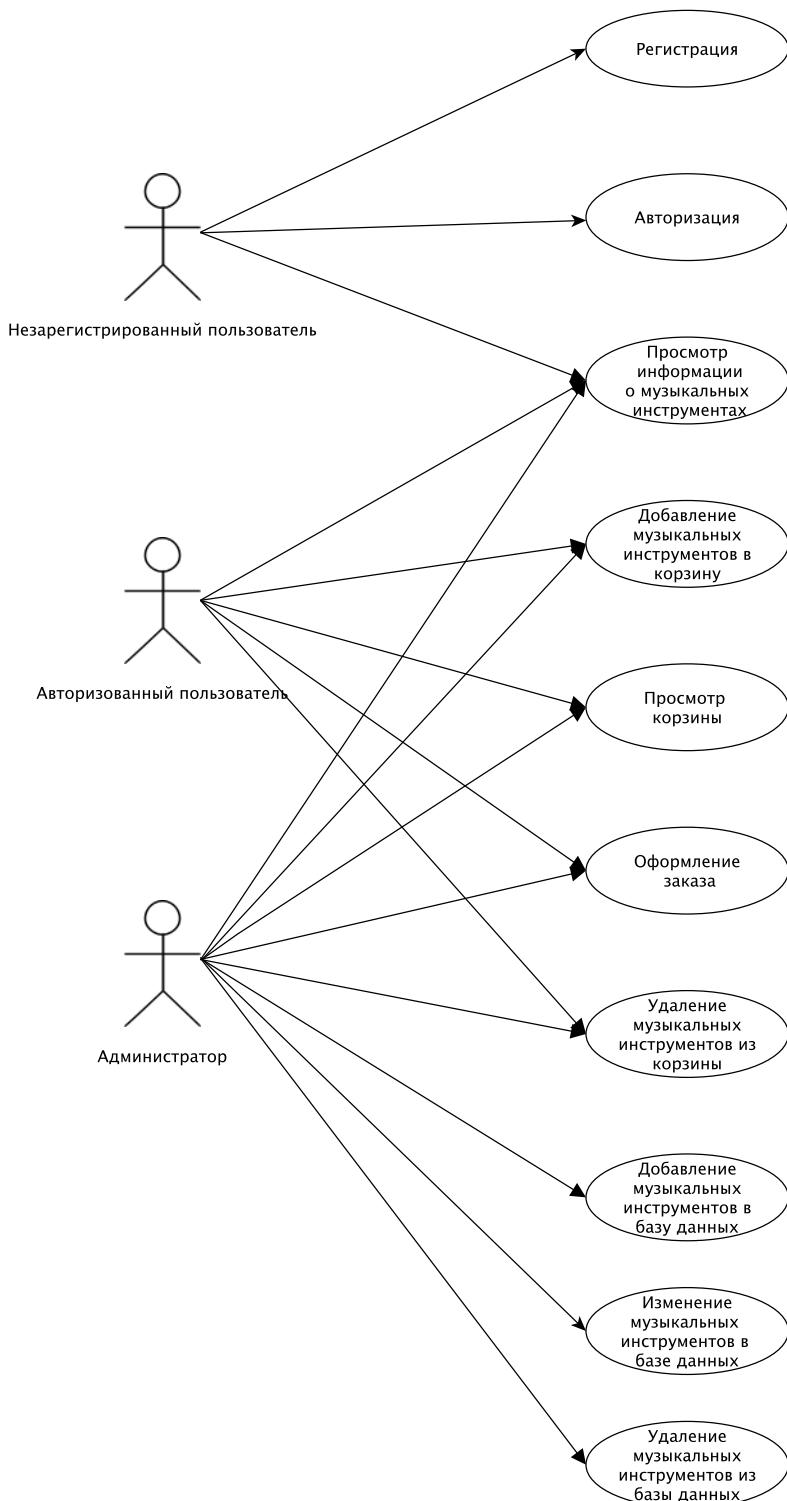


Рисунок 1.1 – Диаграмма использования приложения

## 1.4 Формализация данных

База данных состоит из шести сущностей:

- 1) музикальный инструмент;
- 2) бренд;
- 3) пользователь;
- 4) корзина;
- 5) заказ;
- 6) элемент заказа.

На рисунке 1.2 приведена ER-диаграмма сущностей в нотации Чена.

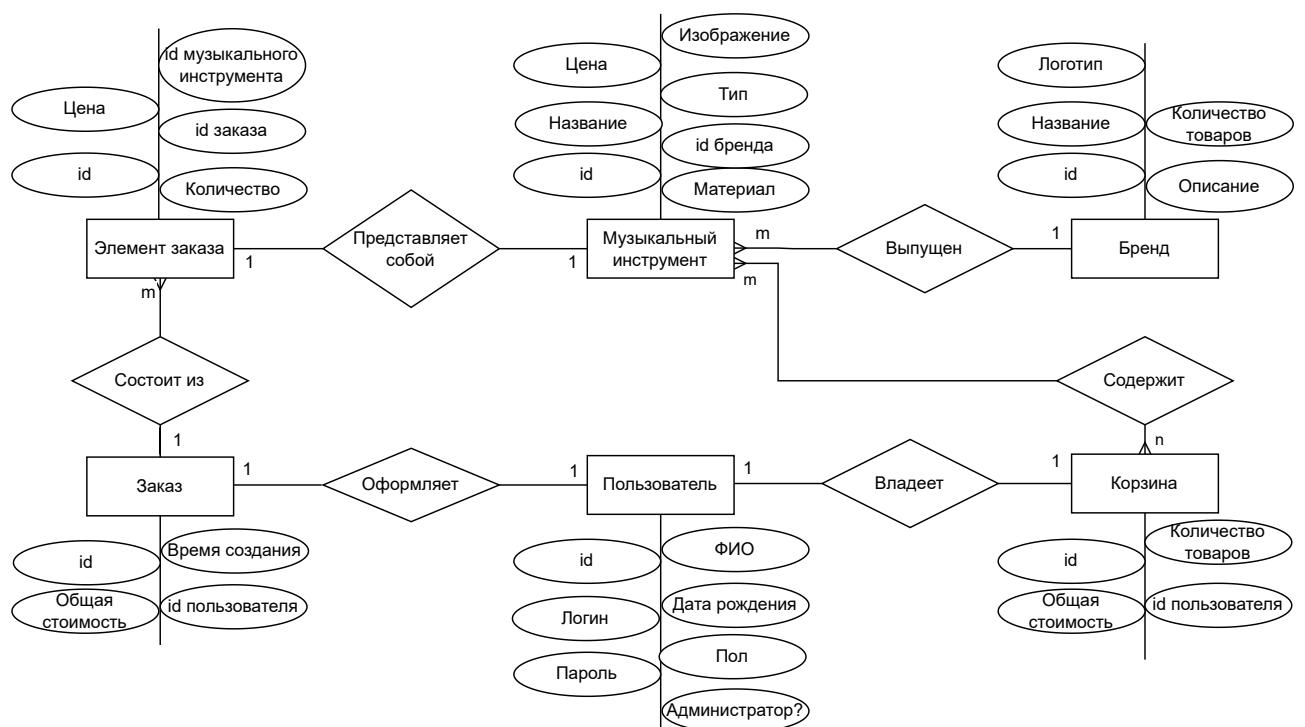


Рисунок 1.2 – ER-диаграмма сущностей в нотации Чена

## 1.5 Анализ моделей данных

Модель данных – это совокупность правил порождения структур данных в базе данных, операций над ними, а также ограничений целостности,

определяющих допустимые связи и значения данных, последовательность их изменения [2].

Рассмотрим основные модели данных.

### **1.5.1 Иерархическая модель данных**

Основополагающей логической структурой для иерархической модели является ориентированное дерево с корнем [3]. Вершины дерева соответствуют интересующим нас объектам, а дуги – связям между объектами. Все вершины дерева, за исключением корня, должны иметь предка. Между двумя вершинами может быть только одна связь. Связи вершины с непосредственно подчиненными вершинами должны иметь определенное упорядочение, как правило, слева направо.

Основными операциями манипулирования данными в иерархической модели являются: поиск указанного экземпляра дерева; переход от одного дерева к другому; переход от одной записи к другой внутри дерева; вставка новой записи в указанную позицию; удаление текущей записи и т. д.

Свойства иерархической модели данных:

- 1) Возможность работы с иерархически упорядоченной информацией.
- 2) Невозможность хранения экземпляров, не имеющих родительских записей.
- 3) Трудность реализации связей «многие ко многим».

Таким образом, иерархическая модель данных не подходит для достижения поставленной цели, потому что формализованные в разделе 1.4 данные имеют связь «многие ко многим», что является проблемой для данной модели.

### **1.5.2 Сетевая модель данных**

Сетевой подход к организации данных является расширением иерархического [3]. В иерархической модели запись-потомок должна иметь в точности одного предка; в сетевой модели запись-потомок может иметь любое число предков. Для реализации иерархической структуры используются две группы типов данных: записи и набор. Набор устанавливает именованную связь для

записи-предка и одной или нескольких записей-потомков, т. е. поддерживает связи «один к одному» и «один ко многим».

Свойства сетевой модели данных:

- 1) Возможность установления произвольных связей между записями.
- 2) Высокая сложность схемы базы данных.

Таким образом, сетевая модель данных также не подходит для достижения поставленной цели, потому что формализованные в разделе 1.4 данные имеют связь «многие ко многим», что является причиной возрастания сложности схемы базы данных при использовании данной модели.

### **1.5.3 Реляционная модель данных**

Реляционная модель основана на математическом понятии отношения, физическим представлением которого является двухмерная таблица, состоящая из строк одинаковой структуры [3]. Логическая структура данных представляется набором связанных таблиц. Модель поддерживает связи «один к одному» и «один ко многим». Связь «многие ко многим» реализуется с помощью декомпозиции.

Свойства реляционной модели данных:

- 1) Независимость от данных.
- 2) Простота организации любой связи данных.
- 3) Ограниченный набор операций.

Таким образом, для решаемой задачи выбрана реляционная модель данных, так как, используя ее, легко организовать связь «многие ко многим», которую образуют данные, формализованные в разделе 1.4.

## **1.6 Анализ систем управления базами данных**

Сравнение систем управления базами данных будет проводится по следующим критериям.

- 1) Доступность в РФ.
- 2) Некоммерческий продукт.

- 3) Поддержка JSON.
- 4) Количество рецензируемых учебных изданий (по оценке [4], [5], [6]).

В таблице 1.2 приведено сравнение систем управления базами данных по заданным выше критериям.

Таблица 1.2 – Сравнение систем управления базами данных

Решение	1	2	3	4
PostgreSQL	+	+	+	7
MySQL	-	+	-	31
Oracle	-	-	+	10
SQL Server	+	-	-	22

Таким образом, в качестве системы управления базами данных для достижения поставленной цели выбрана PostgreSQL.

## 2 Конструкторский раздел

### 2.1 Формализация сущностей системы

На основе выделенных ранее сущностей спроектированы следующие таблицы базы данных.

- 1) Таблица User – содержит информацию о пользователях системы.

Таблица 2.1 – Описание полей таблицы User

Поле	Значение
userId	Уникальный идентификатор пользователя; целочисленный тип, первичный ключ
fio	ФИО пользователя; символьный тип
dateBirth	Дата рождения пользователя; тип даты
login	Логин пользователя; символьный тип, уникальное поле
password	Пароль пользователя; символьный тип
gender	Пол пользователя; символьный тип (ограничение: «Мужской» или «Женский»)
isAdmin	Является ли пользователь системы администратором; логический тип

- 2) Таблица Instrument – содержит информацию о музыкальных инструментах.

Таблица 2.2 – Описание полей таблицы **Instrument**

<b>Поле</b>	<b>Значение</b>
<b>InstrumentId</b>	Уникальный идентификатор музыкального инструмента; целочисленный тип, первичный ключ
<b>name</b>	Название музыкального инструмента; символьный тип;
<b>price</b>	Цена музыкального инструмента; целочисленный тип
<b>material</b>	Материал музыкального инструмента; символьный тип;
<b>type</b>	Тип музыкального инструмента; символьный тип
<b>brandId</b>	Идентификатор бренда музыкальных инструментов; целочисленный тип, в методических целях не является внешним ключем, так как согласно техническому заданию необходимо реализовать триггер, выполняющий функцию каскадного удаления.

3) Таблица **Brand** – содержит информацию о брэндах музыкальных инструментов.

Таблица 2.3 – Описание полей таблицы **Brand**

<b>Поле</b>	<b>Значение</b>
<b>brandId</b>	Уникальный идентификатор бренда; целочисленный тип, первичный ключ
<b>name</b>	Название бренда; символьный тип;
<b>amount</b>	Количество товаров бренда; целочисленный тип
<b>info</b>	Информация о бренде в виде HTML-документа; символьный тип
<b>logo</b>	Ссылка на логотип бренда; символьный тип

4) Таблица **Cart** – содержит информацию о корзинах с товарами.

Таблица 2.4 – Описание полей таблицы Cart

Поле	Значение
cartId	Уникальный идентификатор корзины; целочисленный тип, первичный ключ
amount	Количество товаров в корзине; целочисленный тип
totalPrice	Общая стоимость товаров в корзине; целочисленный тип
userId	Идентификатор пользователя, владельца корзины; целочисленный тип, внешний ключ

- 5) Таблица CartInstrument – развязочная таблица, реализующая связь многие-ко-многим между музыкальными инструментами и корзинами.

Таблица 2.5 – Описание полей таблицы CartInstrument

Поле	Значение
cartInstrumentId	Уникальный идентификатор записи; целочисленный тип, первичный ключ
cartId	Идентификатор корзины; целочисленный тип, внешний ключ
InstrumentId	Идентификатор музыкального инструмента; целочисленный тип, внешний ключ

- 6) Таблица Order – таблица, содержащая информацию о заказах.

Таблица 2.6 – Описание полей таблицы Order

Поле	Значение
orderId	Уникальный идентификатор заказа; целочисленный тип, первичный ключ
price	Общая стоимость заказа; целочисленный тип
time	Время создания заказа; временной тип
userId	Идентификатор пользователя, владельца корзины; целочисленный тип, внешний ключ

- 7) Таблица OrderElement – таблица, содержащая информацию об элементах заказов.

Таблица 2.7 – Описание полей таблицы OrderElement

Поле	Значение
orderElementId	Уникальный идентификатор элемента заказа; целочисленный тип, первичный ключ
price	Стоимость элемента заказа, целочисленный тип
amount	Количество товаров в элементе заказа; целочисленный тип
instrumentId	Идентификатор музыкального инструмента; целочисленный тип, внешний ключ
orderId	Идентификатор заказа; целочисленный тип, внешний ключ

ER-диаграмма сущностей в нотации Мартина представлена на рисунке 2.1.

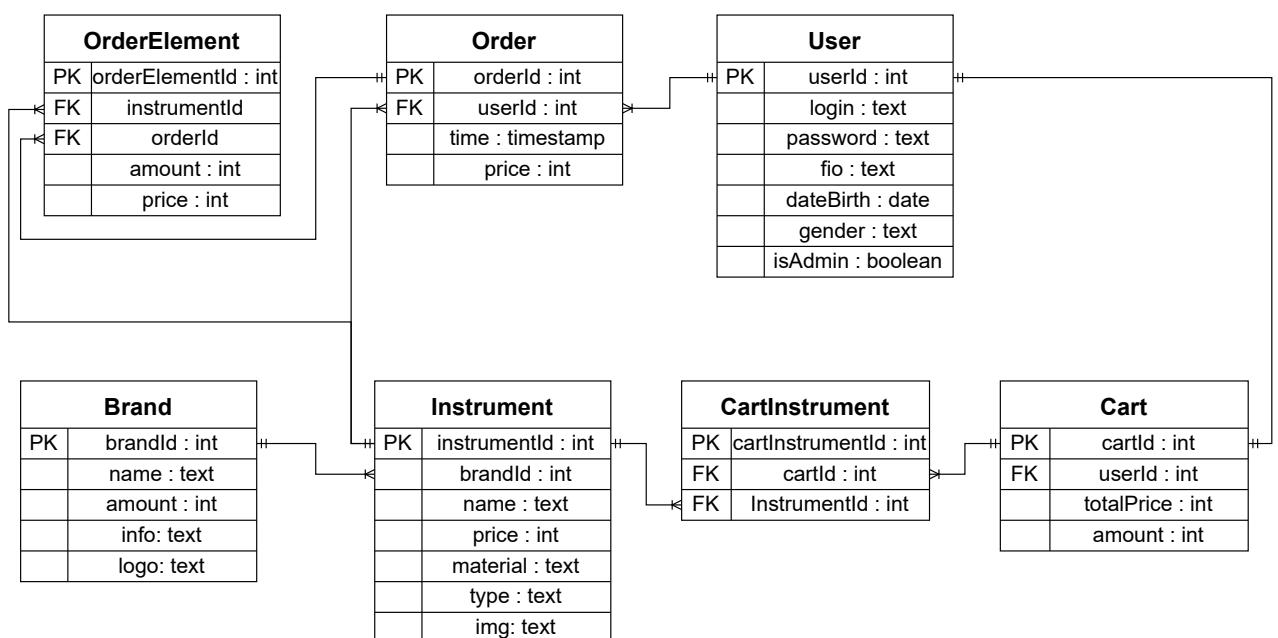


Рисунок 2.1 – ER-диаграмма сущностей в нотации Мартина

## 2.2 Проектирование триггера

В системе должен быть предусмотрен механизм удаления всех музыкальных инструментов удаленного бренда. Механизм представляет собой триггер AFTER на запрос DELETE в таблицу Brand.

Схема алгоритма работы триггера представлена на рисунке 2.2.



Рисунок 2.2 – Схема алгоритма работы триггера

## **2.3 Ролевая модель**

Для работы пользователей с системой на уровне базы данных выделена следующая ролевая модель.

- 1) Admin – администратор. Обладает правами:
  - SELECT/UPDATE/DELETE/INSERT над таблицей Instrument.
- 2) Client – авторизованный пользователь. Обладает правами:
  - SELECT над таблицей Instrument.
  - SELECT над таблицей User.
  - SELECT/UPDATE над таблицей Cart.
  - INSERT над таблицей OrderElement.
  - SELECT/INSERT над таблицей Order.
- 3) Guest – неавторизованный пользователь. Обладает правами:
  - SELECT/INSERT над таблицей User.
  - SELECT над таблицей Instrument.
  - SELECT/UPDATE/INSERT над таблице Cart.

### 3 Технологический раздел

#### 3.1 Разработка сущностей системы

В соответствии с выбранной СУБД, спроектированными базой данных и ее ограничениями целостности разработано создание базы данных, ее сущностей, ограничений целостности и наполнение этой базы данных тестовыми значениями. Реализация представлена в листингах А.1 – А.4 в приложении А.

#### 3.2 Разработка триггера

Был разработан спроектированный триггер AFTER на запрос DELETE в таблицу Brand. Реализация представлена в листинге А.5 в приложении А.

#### 3.3 Разработка ролевой модели

Была разработана спроектированная ролевая модель системы. Реализация представлена в листингах А.6 – А.8 в приложении А.

#### 3.4 Тестирование

Было проведено интеграционное тестирование репозиториев компонента доступа к данным. Для тестов поднималась отдельная копия разработанной базы данных. Тестирование стало частью конвейера на кафедральном Gitlab [7].

На рисунке 3.1 изображено графическое представление всех стадий конвейера.

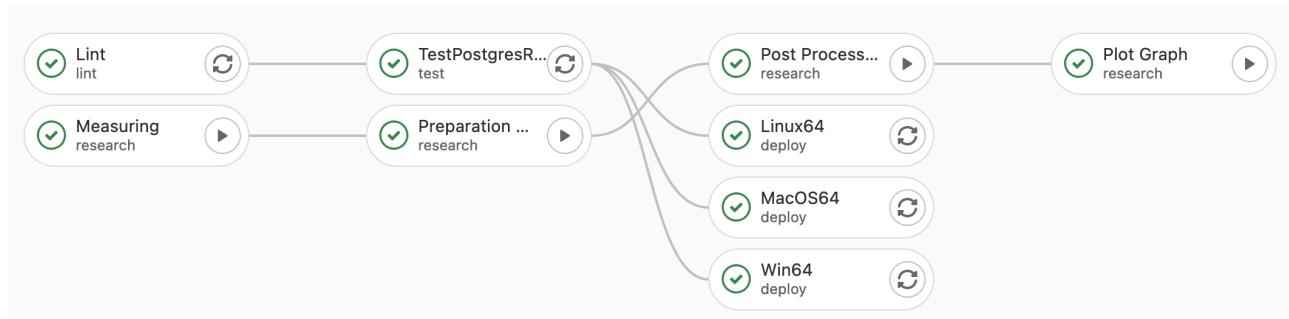


Рисунок 3.1 – Графическое представление всех стадий конвейера

### 3.5 Интерфейс приложения

Для работы с базой данных было разработано приложение, состоящее из двух частей: backend и frontend.

В качестве backend составляющей был разработан http-сервер [8] с чистой архитектурой [9], написанный на языке Go [10].

В качестве frontend составляющей было разработано react-приложение [11] с локальным сервером разработки Vite [12].

На рисунке 3.2 представлен интерфейс домашней страницы неавторизованного пользователя.

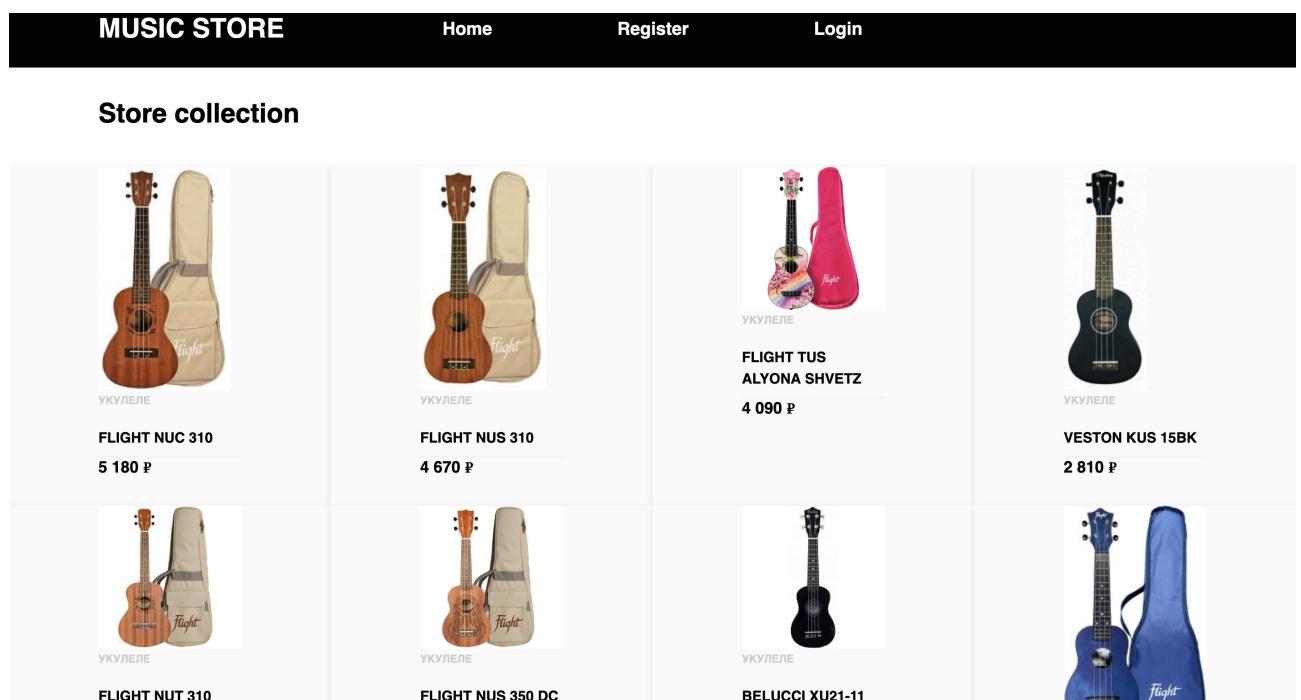
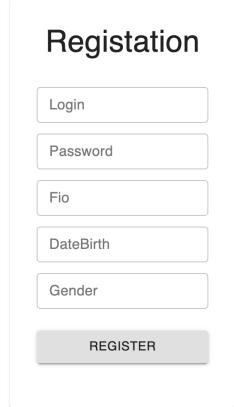


Рисунок 3.2 – Интерфейс домашней страницынеавторизованного пользователя

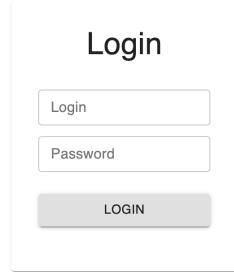
На рисунке 3.3 представлен интерфейс страницы регистрации.



The registration form is titled "Registration". It contains five input fields: "Login", "Password", "Fio", "DateBirth", and "Gender". Below the fields is a "REGISTER" button.

Registration	
Login	
Password	
Fio	
DateBirth	
Gender	
REGISTER	

Рисунок 3.3 – Интерфейс страницы регистрации



The login form is titled "Login". It contains two input fields: "Login" and "Password". Below the fields is a "LOGIN" button.

Login	
Login	
Password	
LOGIN	

Рисунок 3.4 – Интерфейс страницы авторизации

На рисунке 3.5 представлен интерфейс домашней страницы зарегистрированного пользователя.

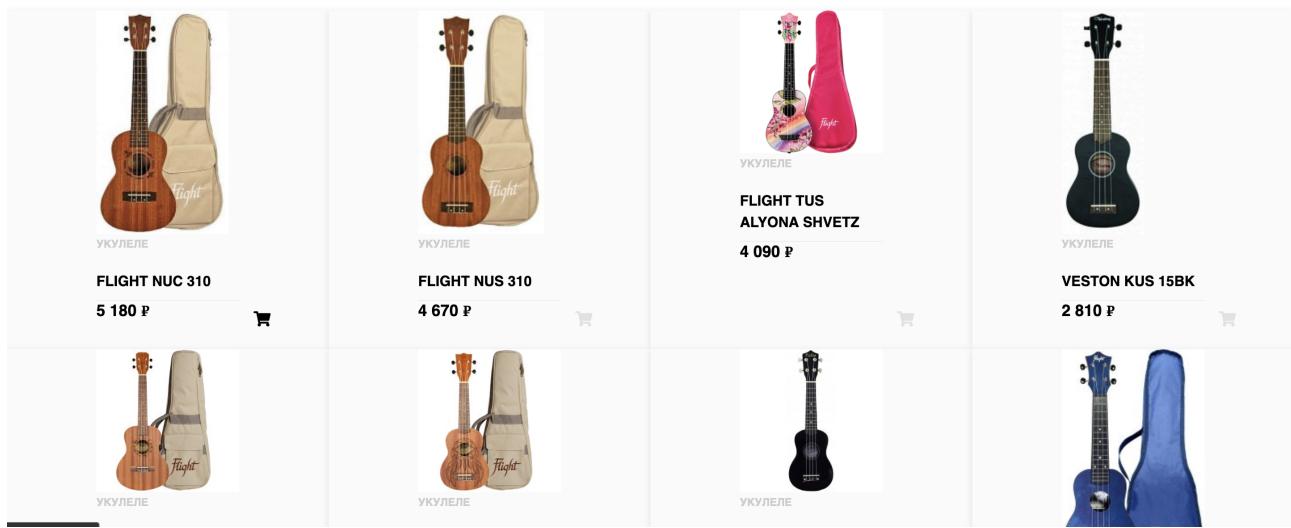
**Store collection**

Рисунок 3.5 – Интерфейс домашней страницы зарегистрированного пользователя

На рисунке 3.6 представлен интерфейс страницы корзины.

**Cart**

Amount: 4

Total Price: 42840

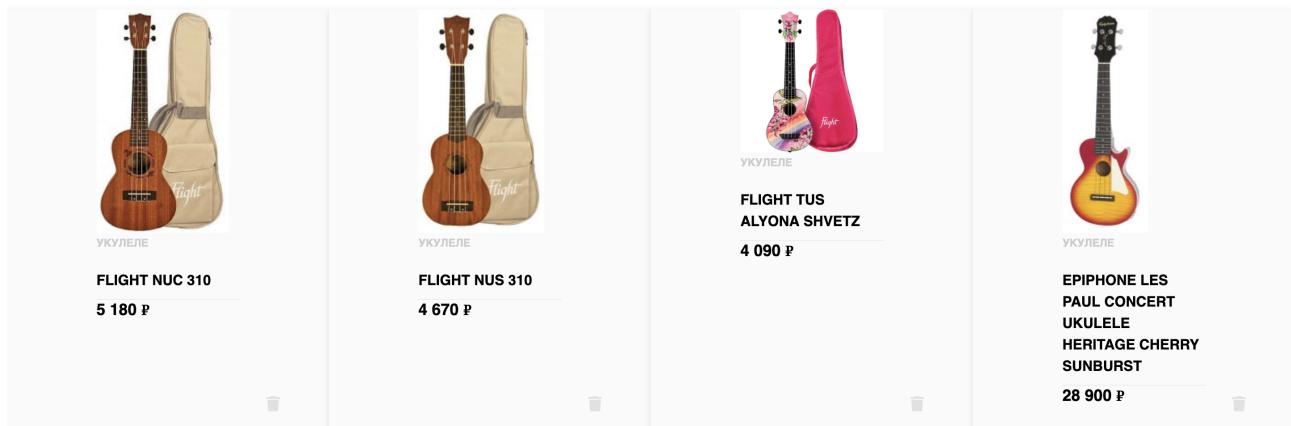
**Checkout**

Рисунок 3.6 – Интерфейс страницы корзины

На рисунке 3.7 представлен интерфейс домашней страницы администра-

топа.

MUSIC STORE

Home Cart Logout

Add instrument in DB

**Store collection**

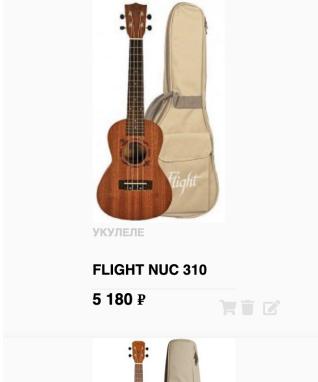
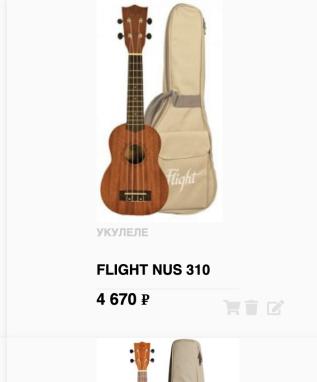
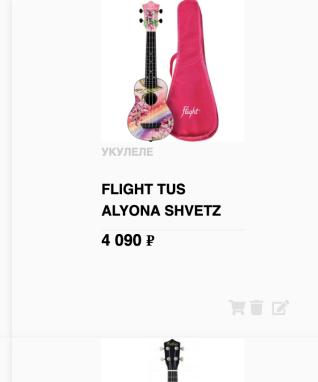
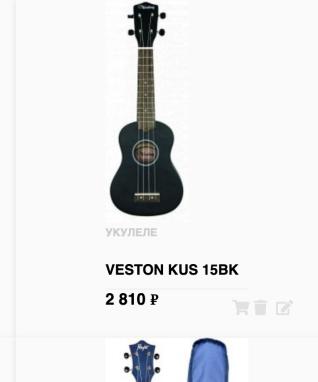
 <p>УКУЛЕЛЕ</p> <p><b>FLIGHT NUC 310</b></p> <p><b>5 180 ₽</b></p> <p> </p>	 <p>УКУЛЕЛЕ</p> <p><b>FLIGHT NUS 310</b></p> <p><b>4 670 ₽</b></p> <p> </p>	 <p>УКУЛЕЛЕ</p> <p><b>FLIGHT TUS ALYONA SHVETZ</b></p> <p><b>4 090 ₽</b></p> <p> </p>	 <p>УКУЛЕЛЕ</p> <p><b>VESTON KUS 15BK</b></p> <p><b>2 810 ₽</b></p> <p> </p>
 <p>УКУЛЕЛЕ</p>	 <p>УКУЛЕЛЕ</p>	 <p>УКУЛЕЛЕ</p>	 <p>УКУЛЕЛЕ</p>

Рисунок 3.7 – Интерфейс домашней страницы администратора

## 4 Исследовательский раздел

### 4.1 Технические характеристики устройства

Технические характеристики устройства, на котором выполнялось исследование, следующие.

- Операционная система macOS Ventura 13.3.1 [13].
- Оперативная память: 8 ГБ.
- Процессор: Apple M1 [14].

### 4.2 Измерение времени выполнения запросов доступа к метаданным

В качестве метаданных была выбрана информация о схемах в базе данных.

Результаты замеров приведены в таблице 4.1. Для таблицы введено обозначение:  $S$  – количество схем в базе данных.

Таблица 4.1 – Таблица времени выполнения запросов доступа к метаданным в МКС.

$S$	Представления информационной схемы	Системные таблицы	Системные хранимые процедуры
0	2652.28	4027.79	4104.01
1	1515.72	3176.53	3032.62
2	1545.91	3031.50	3201.68
3	1523.39	3037.57	3142.70
4	1518.16	3042.14	3254.59
5	1526.45	3108.64	3201.98
6	1564.19	3082.07	3075.74
7	1559.65	3095.35	3048.07
8	1566.30	3076.12	3028.22
9	1649.47	3063.29	3054.24
10	1628.76	3041.30	3054.35

На рисунке 4.1 приведен график, отражающий зависимость времени выполнения запросов от количества схем в базе данных.

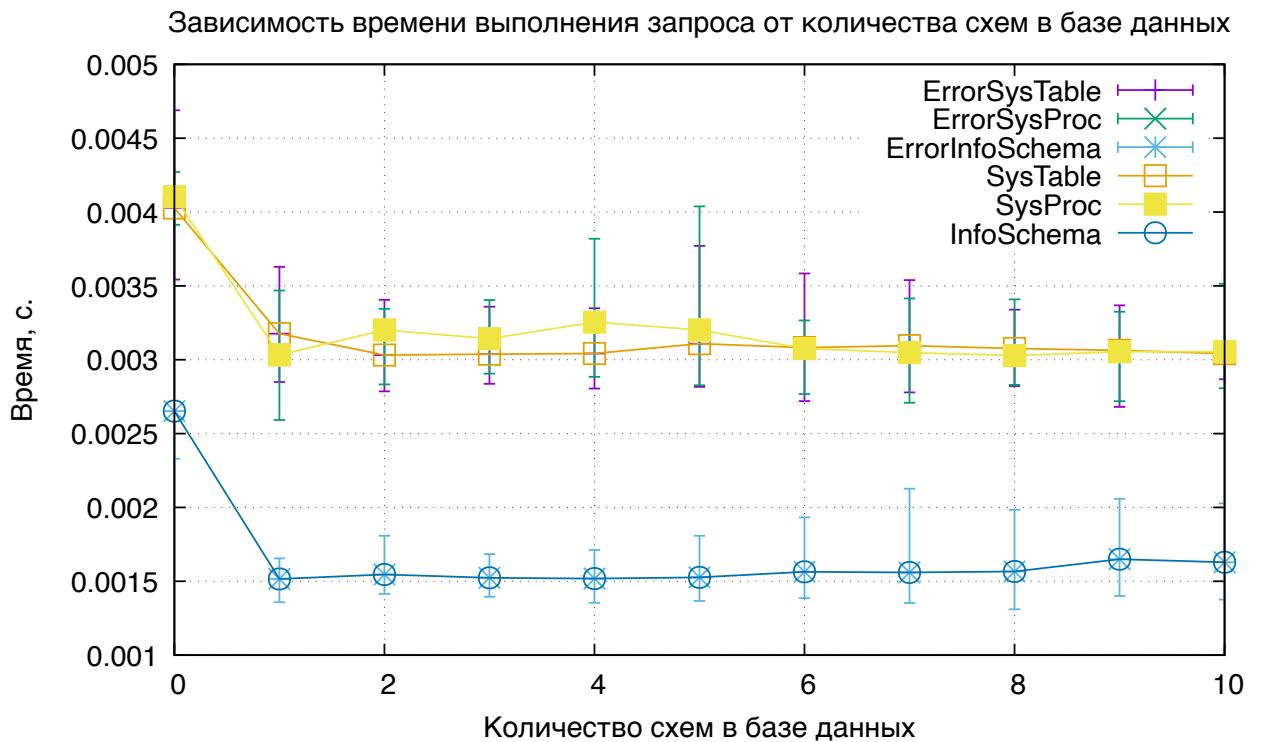


Рисунок 4.1 – Зависимость времени выполнения запросов от количества схем в базе данных

### 4.3 Выводы из исследовательского раздела

Наилучшее время выполнения запроса доступа к метаданным показал метод доступа с помощью представлений информационной схемы. Так, при количестве схем, равном 10, время выполнения запроса с использованием представлений информационной схемы в  $\approx 1.87$  раза быстрее времени выполнения запроса с использованием системных таблиц и в  $\approx 1.88$  раза быстрее времени выполнения запроса с использованием системных хранимых процедур.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы было спроектирована и разработана база данных для цифрового магазина музыкальных инструментов.

В процессе выполнения данной работы были выполнены все задачи:

- 1) Сформулированы требования и ограничения к разрабатываемой базе данных и приложению для цифрового магазина музыкальных инструментов.
- 2) Проведен анализ существующих решений.
- 3) Сформулировано описание пользователей проектируемого приложения для доступа к базе данных.
- 4) Формализованы данные, хранящиеся в базе данных.
- 5) Проведен анализ моделей данных и выбрана подходящая модель.
- 6) Проведен анализ систем управления базами данных и выбрана подходящая система.
- 7) Спроектированы сущности базы данных и ограничения целостности цифрового магазина музыкальных инструментов.
- 8) Спроектирован триггер такой, что если удаляется бренд музыкальных инструментов, то удаляются все товары, связанные с этим брендом.
- 9) Спроектирована ролевую модель на уровне базы данных цифрового магазина музыкальных инструментов.
- 10) Разработаны сущности базы данных цифрового магазина музыкальных инструментов и спроектированные ограничения целостности базы данных.
- 11) Разработан спроектированный триггер такой, что если удаляется бренд музыкальных инструментов, то удаляются все товары, связанные с этим брендом.
- 12) Разработана спроектированная ролевая модель на уровне базы данных цифрового магазина музыкальных инструментов.

- 13) Описаны методы тестирования разработанного функционала и разработаны тесты для проверки корректности работы приложения.
- 14) Описан интерфейс доступа к базе данных.
- 15) Проведен сравнительный анализ методов доступа к метаданным: с помощью системных таблиц, системных хранимых процедур и представлений информационной схемы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Медведева О. С.* Перспективы и возможности онлайн-маркетинга в различной торговле // Вестник Алтайской академии экономики и права. — 2018.
2. *Карпова И. П.* Базы данных. Учебное пособие. — М.: Московский государственный институт электроники и математики (Технический университет), 2009. — 131 с.
3. *Исаченко А. Н.* Модели данных и СУБД. — Минск: БГУ, 2007.
4. Читай-город [Электронный ресурс]. — Режим доступа: <https://www.chitai-gorod.ru/> (дата обращения: 11.03.2023).
5. Лабиринт [Электронный ресурс]. — Режим доступа: <https://www.labirint.ru/> (дата обращения: 11.03.2023).
6. Book24 [Электронный ресурс]. — Режим доступа: <https://book24.ru/> (дата обращения: 11.03.2023).
7. Gitlab ИУ7 [Электронный ресурс]. — Режим доступа: <https://git.iu7.bmstu.ru/> (дата обращения: 11.03.2023).
8. *Бондаренко Т., Федотов Е., Бондаренко А.* Разработка HTTP сервера // Инженерный вестник Дона. — 2018. — 2 (49). — С. 115.
9. *Мартин Р.* Чистая архитектура. Искусство разработки программного обеспечения. — Издательский дом, Санкт-Петербург, 2018.
10. Язык программирования Go [Электронный ресурс]. — Режим доступа: <https://go.dev/> (дата обращения: 11.03.2023).
11. *Попков И., Курзаева Л.* Использование React для разработки веб-приложений // Аллея науки. — 2018. — Т. 1, № 7. — С. 924—927.
12. Локальный сервер разработки Vite [Электронный ресурс]. — Режим доступа: <https://vitejs.dev/> (дата обращения: 11.03.2023).
13. macOS Ventura 13.3.1 [Электронный ресурс]. — Режим доступа: <https://support.apple.com/ru-ru/HT213721> (дата обращения: 11.03.2023).
14. Apple M1 [Электронный ресурс]. — Режим доступа: <https://www.apple.com/macbook-air-m1/> (дата обращения: 11.03.2023).

# ПРИЛОЖЕНИЕ А

Листинг А.1 – Создание базы данных и ее сущностей (начало)

```
1 drop schema if exists store cascade;
2
3 create schema store;
4
5 create table store.Brands
6 (
7     brand_id serial primary key,
8     brand_name text not null,
9     brand_amount int,
10    brand_info text,
11    brand_logo text not null
12 );
13
14 create table store.Instruments
15 (
16     instrument_id serial primary key,
17     instrument_name text not null,
18     instrument_price int,
19     instrument_material text not null,
20     instrument_type text not null,
21     instrument_img text not null,
22     brand_id int not null
23 );
24
25 create table store.Users
26 (
27     user_id serial primary key,
28     user_login text not null,
29     user_password text not null,
30     user_fio text not null,
31     user_date_birth date,
32     user_gender text not null,
33     user_is_admin boolean
34 );
35
36 create table store.Orders
37 (
```

Листинг А.2 – Реализация создания базы данных и ее сущностей (окончание)

```
1  order_id serial primary key,  
2  order_time date,  
3  order_price int,  
4  user_id int not null,  
5  foreign key (user_id) references store.Users(user_id) on delete  
6    cascade  
7 );  
8  
9 create table store.Order_elements  
10 ( 11  
12   order_element_id serial primary key,  
13   order_element_amount int,  
14   order_element_price int,  
15   instrument_id int not null,  
16   order_id int not null,  
17   foreign key (instrument_id) references store.Instruments(  
18     instrument_id) on delete cascade,  
19   foreign key (order_id) references store.Orders(order_id) on  
20     delete cascade  
21 );  
22  
23 create table store.Carts  
24 ( 25  
26   cart_id serial primary key,  
27   user_id int not null,  
28   foreign key (user_id) references store.Users(user_id) on delete  
29     cascade,  
30   cart_total_price int,  
31   cart_amount int  
32 );  
33  
34 create table store.carts_instruments ( 35  
35   carts_instruments_id serial primary key,  
36   cart_id int not null,  
37   instrument_id int not null,  
38   foreign key (cart_id) references store.Carts(cart_id) on delete  
39     cascade,  
40   foreign key (instrument_id) references store.Instruments(  
41     instrument_id) on delete cascade  
42 );
```

Листинг А.3 – Реализация создания ограничений целостности базы данных

```
1 alter table store.Users  
2   add constraint correct_user_gender CHECK (user_gender = 'Male' OR  
3                                             user_gender = 'Female');
```

Листинг А.4 – Реализация наполнения базы данных тестовыми значениями

```
1 copy store.Brands FROM '/var/lib/postgresql/data/brands.csv'  
    DELIMITER ';' CSV HEADER;  
2 copy store.Instruments FROM '/var/lib/postgresql/data/instruments.  
    csv' DELIMITER ';' CSV HEADER;  
3 copy store.Users FROM '/var/lib/postgresql/data/users.csv'  
    DELIMITER ';' CSV HEADER;  
4 copy store.Carts FROM '/var/lib/postgresql/data/carts.csv'  
    DELIMITER ';' CSV HEADER;
```

Листинг А.5 – Реализация создания триггера AFTER на запрос Delete в таблицу Brand

```
1 create or replace function delete_instruments_after_deleting_brand  
    ()  
2 returns trigger as  
3 $$ begin  
4     delete from store.instruments i where i.brand_id = old.brand_id;  
5     return old;  
6 end; $$  
7 language plpgsql;  
8  
9 create trigger delete_instruments_after_deleting_brand after delete  
    on store.brands for each row execute procedure  
    delete_instruments_after_deleting_brand();
```

Листинг А.6 – Реализация ролевой модели системы (начало)

```
1 create or replace function store.create_role (r_name text, r_sql  
    text)  
2 returns void as  
3 $$  
4 begin  
5     if not exists (select from pg_catalog.pg_roles  
                      where rolname = r_name) then  
          execute r_sql;  
6     end if;  
7 end; $$
```

Листинг А.7 – Реализация ролевой модели системы (продолжение)

```
1 language 'plpgsql';
2
3 create or replace function store.delete_role (r_name text)
4 returns void as
5 $$
6 begin
7   if exists (select from pg_catalog.pg_roles
8     where rolname = r_name) then
9     execute format( 'reassign owned by %s to postgres;' ||
10                   'drop owned by %s;' ||
11                   'drop role %s;', r_name, r_name, r_name);
12   end if;
13 end;
14 $$
15 language 'plpgsql';
16 select store.delete_role('guest');
17 select store.create_role('guest', 'create role guest LOGIN PASSWORD
18   ''guest'';');
19 grant select, insert on store.users to guest;
20 grant select on store.instruments to guest;
21 grant update on store.users_user_id_seq to guest;
22 grant update on store.carts_cart_id_seq to guest;
23 grant select, update, insert on store.carts to guest;
24
25 select store.delete_role('client');
26 select store.create_role('client', 'create role client LOGIN
27   PASSWORD ''client'';');
28 grant select on store.instruments to client;
29 grant select on store.users to client;
30 grant select, update on store.carts to client;
31 grant select, delete, insert on store.carts_instruments to client;
32 grant update on store.carts_instruments_carts_instruments_id_seq to
33   client;
34 grant update on store.order_elements_order_element_id_seq to client
35   ;
36 grant insert on store.order_elements to client;
37 grant update on store.orders_order_id_seq to client;
38 grant select, insert on store.orders to client;
39
40 select store.delete_role('admin');
```

Листинг А.8 – Реализация ролевой модели системы (окончание)

```
1 select store.create_role('admin', 'create role admin LOGIN PASSWORD
  ''admin'';');
2 grant update on store.instruments_instrument_id_seq to admin;
3 grant select, insert, update, delete on store.instruments to admin;
4
5 grant all on schema store to guest;
6 grant all on schema store to client;
7 grant all on schema store to admin;
```