



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»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### *К КУРСОВОЙ РАБОТЕ*

### *НА ТЕМУ:*

*«Разработка системы для составления нормативных документов рабочей программы дисциплины (РПД) и фонда оценочных средств (ФОС)»*

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

**Ю. В. Строганов**

\_\_\_\_\_  
(И.О. Фамилия)

Консультант

\_\_\_\_\_  
(Подпись, дата)

**О. В. Яковлева**

\_\_\_\_\_  
(И.О. Фамилия)

Студент ИУ7-65Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

**А. Е. Богаченко**

\_\_\_\_\_  
(И.О. Фамилия)

2022 г.

## РЕФЕРАТ

Расчетно-пояснительная записка 37 с., 9 рис., 1 табл., 39 ист., 4 прил.

В работе представлены проектирование и реализация системы для создания нормативных документов рабочей программы дисциплины и фонда оценочных средств, состоящей из приложения и базы данных.

Проведён анализ предметной области, рассмотрены существующие решения для хранения и обработки данных, их особенности и ограничения области применимости, проведено исследование устойчивости выбранного решения к высоким нагрузкам.

### КЛЮЧЕВЫЕ СЛОВА

*рабочая программа дисциплины, модель сущность-связь, фонд оценочных средств, Datomic, нагрузочное тестирование, функциональные языки программирования.*

## СОДЕРЖАНИЕ

<b>Введение</b>	<b>5</b>
<b>1 Аналитическая часть</b>	<b>7</b>
1.1 Анализ предметной области . . . . .	7
1.2 Структура рабочей программы дисциплины . . . . .	8
1.3 Базы данных и системы управления базами данных . . . . .	10
1.3.1 Классификация реляционных баз данных по способу хранения . . . . .	10
1.4 Кэширование данных . . . . .	11
1.4.1 Проблемы кэширования данных . . . . .	11
<b>2 Конструкторская часть</b>	<b>13</b>
2.1 Проектирование системы . . . . .	13
2.1.1 Проектирование базы данных . . . . .	15
<b>3 Технологическая часть</b>	<b>19</b>
3.1 Средства реализации программного обеспечения . . . . .	19
3.2 Выбор базы данных и системы управления базой данных . . . . .	19
3.2.1 Обзор СУБД с строчным хранением . . . . .	19
3.2.2 Обзор NoSQL СУБД . . . . .	20
3.3 Листинги кода . . . . .	22
3.3.1 Создание схемы базы данных . . . . .	23
3.3.2 Доступ к атрибутам сущности базы данных . . . . .	23
3.4 Хранимые процедуры . . . . .	24
3.5 Взаимодействие с приложением . . . . .	24
<b>4 Исследовательская часть</b>	<b>25</b>
4.1 Технические характеристики . . . . .	25

4.2 Тестирование отказоустойчивости . . . . .	25
<b>Заключение</b>	<b>28</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>29</b>
<b>ПРИЛОЖЕНИЕ А</b>	<b>34</b>
<b>ПРИЛОЖЕНИЕ Б</b>	<b>35</b>
<b>ПРИЛОЖЕНИЕ В</b>	<b>36</b>
<b>ПРИЛОЖЕНИЕ Г</b>	<b>37</b>

## Введение

Рабочая программа дисциплины – программа освоения учебного материала, соответствующая требованиям государственного образовательного стандарта высшего профессионального образования и учитывающая специфику подготовки студентов по выбранному направлению или специальности. Разрабатывается для каждой дисциплины учебного плана всех реализуемых в университете основных образовательных программ [1].

Фонд оценочных средств – совокупность оценочных материалов, а также описание форм и процедур, предназначенных для определения уровня достижения обучающимися установленных результатов обучения [2].

Функции хранения, обработки и анализа информации, находящейся в рабочей программе дисциплины, могут быть реализованы в системах управления обучением (англ. Learning Management System (LMS) [3]). Такой интерфейс может предоставить пользователю системы (в данном случае преподавателю) возможность получать и редактировать информацию о рабочей программе дисциплины.

Рабочая программа дисциплины представлена в виде документа формата Microsoft Word [4], что накладывает ограничения на автоматизированную программную обработку и анализ информации, представленной в рабочей программе дисциплины.

Цель работы – спроектировать базу данных для хранения компонентов нормативных документов и реализовать программное обеспечение для их создания, редактирования, удаления и составления на их основе рабочей программы дисциплины и фонда оценочных средств.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- провести анализ предметной области и готовых решений в области баз данных и систем управления базами данных;

- спроектировать базу данных, описать её компоненты и связи;
- реализовать базу данных и программное обеспечение для работы с ней;
- провести исследование устойчивости реализованной системы к высоким нагрузкам.

## **1 Аналитическая часть**

В данном разделе проведён анализ предметной области, описана структура рабочей программы дисциплины, проведена классификация баз данных (БД) и систем управления базами данных (СУБД). Рассмотрен механизм кэшированных данных, описаны возможные проблемы при его использовании и представлены методы их решения.

### **1.1 Анализ предметной области**

Каждая дисциплина, преподаваемая в высшем учебном заведении, имеет свою рабочую программу. В ней хранится различная информация о дисциплине: стандарт, содержание, учебные работы, результаты обучения, перечень литературы, методические указания и прочее. Зачастую, у пользователей нет никаких автоматизированных инструментов для анализа и редактирования таких программ.

Дисциплина имеет свой федеральный государственный образовательный стандарт (ФГОС): 3+, 3++ и другие [5]. Образовательный стандарт – совокупность обязательных требований к образованию определённого уровня и (или) к профессии, специальности. Каждый образовательный стандарт для каждого направления подготовки обучаемого имеет свою компетенцию. Компетенция содержит информацию о том, что должен знать, уметь и какими навыками должен обладать выпускник, успешно освоивший дисциплину.

Каждому направлению подготовки, но одной и той же рабочей программы дисциплины, сопоставлены компетенции, различные в каждом образовательном стандарте. Коды компетенций для каждого образовательного стандарта отличаются.

Далее будет рассмотрена рабочая программа дисциплины «Информатика», соответствующая образовательному стандарту 3++, разработанная и преподаваемая в МГТУ им. Н. Э. Баумана [6].

## **1.2 Структура рабочей программы дисциплины**

Структура файлов рабочих программ дисциплин для каждого вуза может отличаться, но внутри одного вуза, скорее всего, программы имеют одну и ту же (или схожую) структуру. Рабочая программа дисциплины «Информатика» содержит следующие разделы:

- 1) титульный лист;
- 2) планируемые результаты обучения по дисциплине, соотнесённые с планируемыми результатами освоения образовательной программы;
- 3) место дисциплины в структуре образовательной программы;
- 4) объём дисциплины;
- 5) содержание дисциплины;
- 6) учебно-методическое обеспечение самостоятельной работы;
- 7) фонд оценочных средств для проведения текущего контроля и промежуточной аттестации студентов;
- 8) перечень основной и дополнительной литературы;
- 9) методические указания;
- 10) перечень информационных технологий;
- 11) описание материально-технической базы.

Разделы №2, №4, №5 представлены в виде совокупности текстовой информации и таблиц (рис. 1), остальные разделы представлены в виде текстовой информации (рис. 2).



### 3. ОБЪЕМ ДИСЦИПЛИНЫ

Общий объем дисциплины составляет 7 зачетных единиц(з.е.), 252 академических часа.

В том числе:

1 семестр – 4 з.е. (144 ак.ч.),

2 семестр – 3 з.е. (108 ак.ч.).

**Таблица 2.** Объём дисциплины по видам учебных занятий (в академических часах)

Виды учебной работы	Объем по семестрам, ч		
	Всего	Семестры	
		1	2
Объем дисциплины	252	144	108
<b>Аудиторная работа</b>	<b>136</b>	<b>85</b>	<b>51</b>
Лекции (Л)	17	17	0
Семинары (С)	51	34	17
Лабораторные работы (ЛР)	68	34	34
<b>Самостоятельная работа (СР)</b>	<b>116</b>	<b>59</b>	<b>57</b>
Проработка учебного материала лекций	2	2	0
Подготовка к семинарам	6.25	4.25	2
Подготовка к лабораторным работам	50	16	34
Подготовка к экзамену	30	30	0
Подготовка к рубежному контролю	18	6	12
Другие виды самостоятельной работы	9.75	0.75	9
<b>Вид промежуточной аттестации</b>		<b>Экзамен</b>	<b>Зачёт</b>

Рисунок 1 – Изображение таблицы с информацией о объёме дисциплины.

Первый раздел содержит общую информацию о курсе: название, образовательный стандарт и прочее.

## **2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

Дисциплина входит в блок Б1 «Дисциплины (модули)» образовательных программ бакалавриата по направлениям 11.03.04 «Электроника и наноэлектроника», 12.03.02 «Опtotехника», 12.03.05 «Лазерная техника и лазерные технологии», 13.03.02 «Электроэнергетика и электротехника», 13.03.03 «Энергетическое машиностроение», 14.03.01 «Ядерная энергетика и теплофизика», 24.03.01 «Ракетные комплексы и космонавтика», 28.03.02 «Наноинженерия».

Изучение дисциплины предполагает предварительное освоение следующих дисциплин (в рамках школьного курса):

- Основы информатики;
- Математика;
- Иностранный язык (английский язык).

Освоение данной дисциплины необходимо как предшествующее для следующих дисциплин образовательной программы:

- Инженерная и компьютерная графика

Освоение учебной дисциплины связано с формированием компетенций с учетом матриц компетенций ОПОП для направлений (уровень бакалавриата): 11.03.04 «Электроника и наноэлектроника», 12.03.02 «Опtotехника», 12.03.05 «Лазерная техника и лазерные технологии», 13.03.02 «Электроэнергетика и электротехника», 13.03.03 «Энергетическое машиностроение», 14.03.01 «Ядерная энергетика и теплофизика», 24.03.01 «Ракетные комплексы и космонавтика», 28.03.02 «Наноинженерия».

Рисунок 2 – Изображение текстовой информации о месте дисциплины в структуре образовательной программы.

Раздел №2 содержит коды и описания компетенций для каждого направления подготовки. В разделах №4 и №5 хранится информация о нагрузке и структуре дисциплины.

### **1.3 Базы данных и системы управления базами данных**

Для персистентного хранения данных используются базы данных [7], контролируемые системами управления базами данных [8].

#### **1.3.1 Классификация реляционных баз данных по способу хранения**

Реляционные базы данных по способу хранения делятся на две группы: строковые и колоночные.

## **Строковые базы данных**

Строковыми базами данных называются такие базы данных, записи которых хранятся построчно, в основном такой тип используется в транзакционных системах (англ. OLTP [9]), для которых характерно большое количество коротких транзакций с операциями вставки, обновления и удаления данных. Системы OLTP наиболее подходят для быстрой обработки запросов и поддержания целостности данных в средах с множественным доступом.

## **Колоночные базы данных**

Колоночными называются базы данных, записи которых хранятся по столбцам, в основном такой тип используется в аналитических системах (англ. OLAP [10]), которые характеризуются низким объёмом транзакций, а запросы часто являются составными и включают в себя агрегацию, мерой эффективности таких систем является время отклика. Наличие агрегированных и исторических данных позволяет использовать OLAP-системы для реализации методов интеллектуального анализа.

### **1.4 Кэширование данных**

Чтобы уменьшить время отклика, можно использовать механизм кэширования данных, для реализации которого подходят NoSQL [11] in-memory базы данных, хранящие данные в оперативной памяти, что обеспечивает более быстрый доступ, по сравнению с традиционными реализациями.

#### **1.4.1 Проблемы кэширования данных**

##### **Проблема синхронизации данных**

Приложение пишет в кэш и базу данных, которые между собой никак не синхронизируются, таким образом возникает несогласованность данных. Например, возможна ситуация, когда данные удаляются из хранилища и их нужно удалить из кэша. Эту проблему можно решить установкой триггеров, которые будут срабатывать на изменение и удаление записей, чтобы синхронизировать данные в кэше.

## **Проблема «холодного старта»**

Когда кэш только создаётся, он пуст и не имеет никаких данных – все запросы идут напрямую в БД, и только через некоторое время кэш будет содержать необходимую информацию. Эту проблему можно решить, выбрав СУБД с журналированием всех операций – при перезагрузке можно восстановить предыдущее состояние кэша с помощью журнала событий, который хранится на диске. При перезагрузке кэша нужно синхронизировать данные с хранилищем – возможно, какие-то данные, находящиеся в кэше, перестали быть актуальными за время его перезагрузки.

## 2 Конструкторская часть

В данном разделе рассмотрено проектирование базы данных и системы создания нормативных документов рабочей программы дисциплины и фонда оценочных средств, являющейся частью системы управления обучением.

### 2.1 Проектирование системы

Проектируемая система состоит из БД компонентов нормативных документов и пользовательского приложения для их добавления, редактирования и удаления, а также создания РПД и ФОС. Диаграмма последовательностей представлена на рисунке 3.

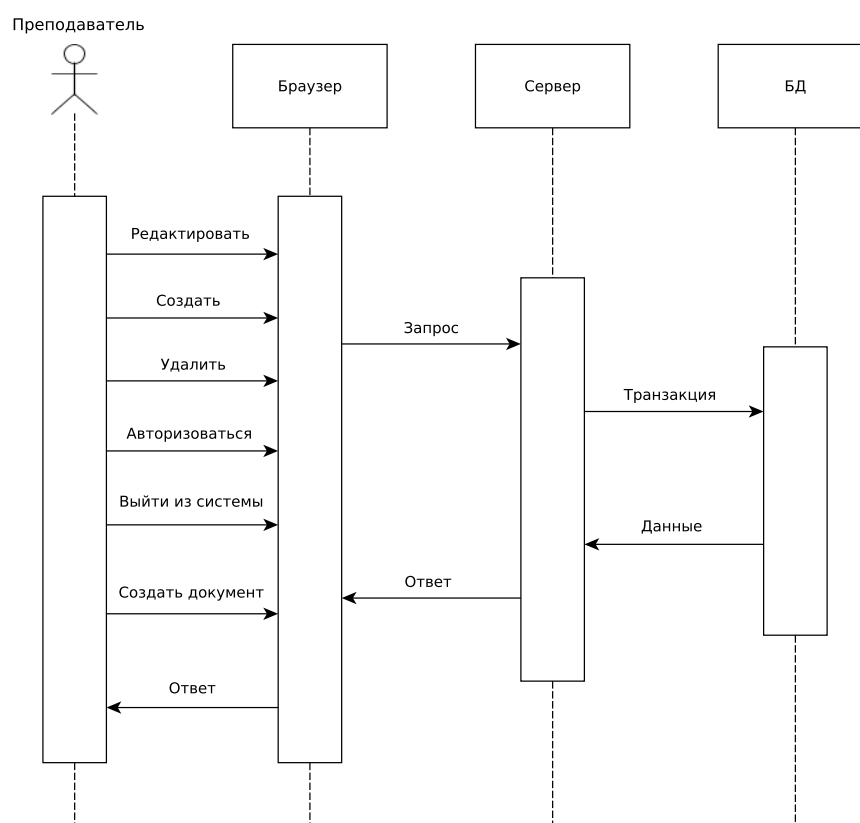


Рисунок 3 – Диаграмма последовательностей

## Роли пользователей

В приложении определены две основные роли:

- 1) преподаватель – работает с компонентами документов, хранящимися в базе данных, и составляет на их основе нормативные документы;
- 2) администратор – осуществляет управление системой.

Схема распределения ролей представлена на рисунке 4.

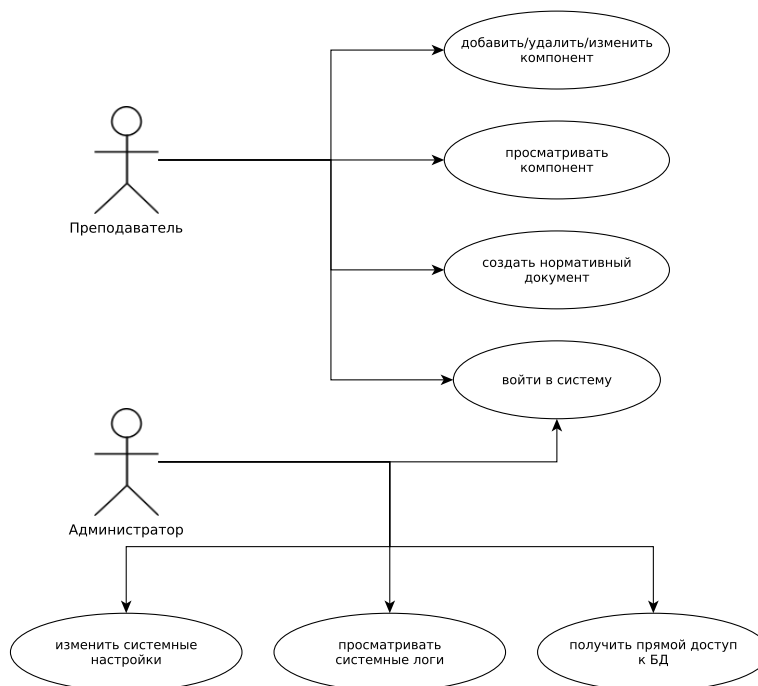


Рисунок 4 – Роли пользователей в системе

## Диаграмма функционирования системы

Верхнеуровневая диаграмма представлена на рисунке 5, процесс составления документа представлен на рисунке 6. На вход система получает данные о компонентах нормативного документа, запрашивает их из базы данных, обрабатывает согласно структуре нормативного документа и на выходе формирует нормативный документ.

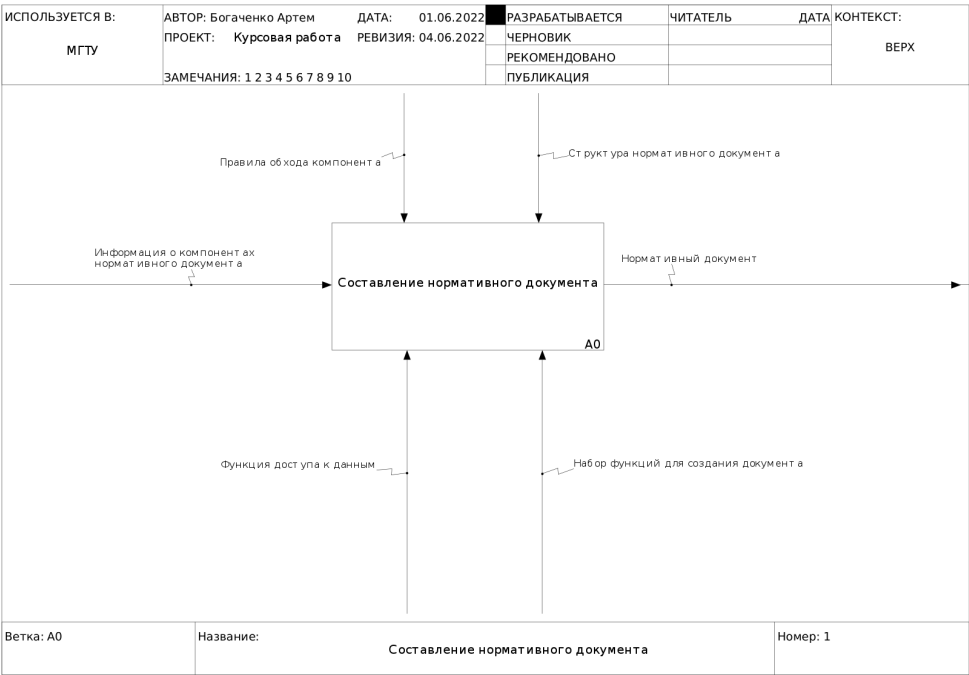


Рисунок 5 – Верхнеуровневая диаграмма приложения

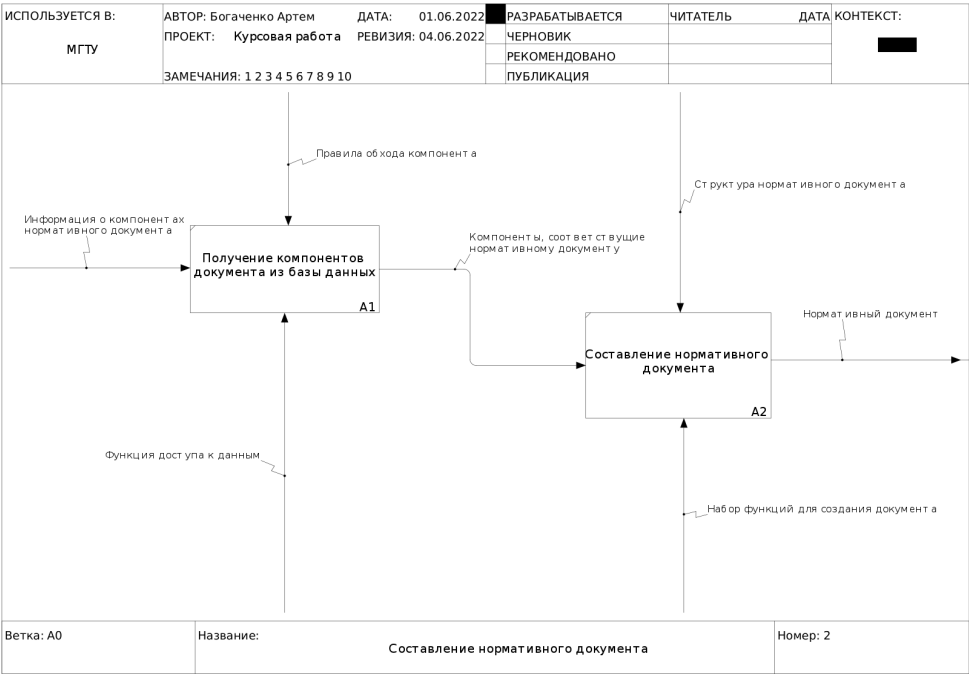


Рисунок 6 – Процесс составления документа

2.1.1 Проектирование базы данных

Модель вида «сущность-связь» [12] представлена на рисунке 7.

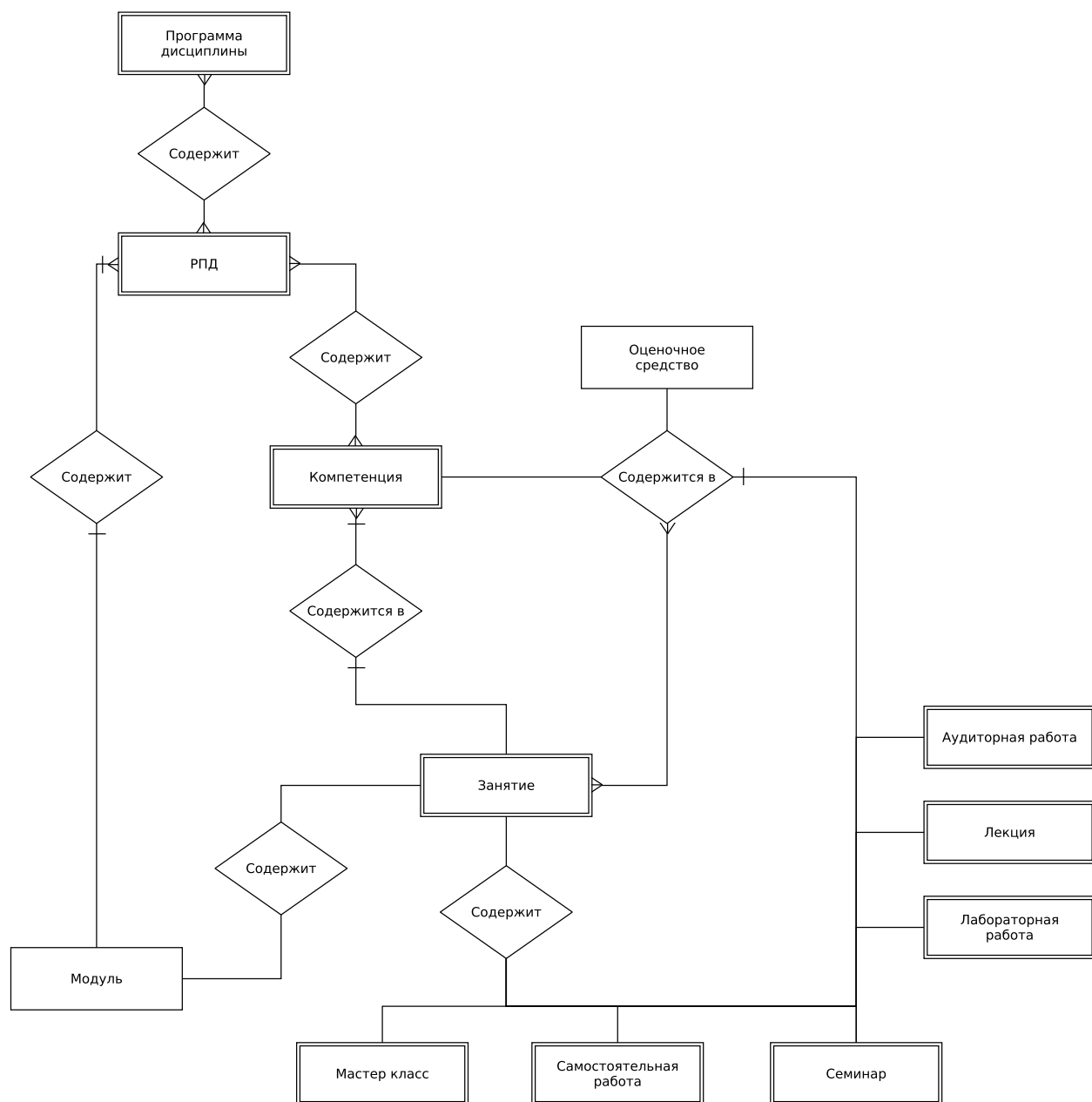


Рисунок 7 – Модель «сущность-связь» в нотации Чена

### Описание сущностей модели

Программа дисциплины имеет следующие атрибуты:

- название;
- описание;
- РПД.



РПД имеет следующие атрибуты:

- название;
- код ФГОС;
- содержание;
- виды учебных занятий;
- компетенции;
- модули;
- результат обучения;
- материалы дисциплины;

Модуль имеет следующие атрибуты:

- занятия;
- компетенции;
- оценочное средство;
- имя;
- количество часов для каждого типа работы;
- номер семестра;

Компетенция имеет следующие атрибуты:

- имя;
- код;
- формулировка;
- тип;
- категория;
- этапы формирования;
- оценочное средство.

Занятие имеет следующие атрибуты:

- имя;
- количество часов;
- компетенция;
- оценочное средство;
- тип занятия:
  - мастер класс;
  - самостоятельная работа;
  - семинар;
  - лабораторная работа;
  - лекция;
  - аудиторная работа.

Мастер класс, самостоятельная работа, семинар, лабораторная работа, лекция и аудиторная работа имеют следующие атрибуты:

- название;
- количество часов;
- тема;
- вид работы;
- оценочное средство.

### **3 Технологическая часть**

В этом разделе проведён выбор средств реализации программного обеспечения, обусловлен выбор базы данных и СУБД, приведены листинги кода, реализующие необходимую для работы системы функциональность.

#### **3.1 Средства реализации программного обеспечения**

В качестве языка программирования был выбран функциональный язык Clojure [14], поддерживающий библиотеки JVM [39], подходящий для обработки больших объёмов данных и построения высоконагруженных систем, а также совместимый с ним функциональный язык ClojureScript [15], адаптированный для работы в браузере.

#### **3.2 Выбор базы данных и системы управления базой данных**

В данном подразделе рассмотрены популярные строчные СУБД, которые могут быть использованы для реализации хранения данных.

##### **3.2.1 Обзор СУБД с строчным хранением**

##### **PostgreSQL**

PostgreSQL [16] – это свободно распространяемая объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных [17].

PostgreSQL предоставляет транзакции со свойствами атомарности, согласованности, изоляции, долговечности (ACID [18]), автоматически обновляемые представления, материализованные представления, триггеры, внешние ключи и хранимые процедуры. Данная СУБД предназначена для обработки ряда рабочих нагрузок, от отдельных компьютеров до хранилищ данных или веб-сервисов с множеством одновременных пользователей.

Рассматриваемая СУБД управляет параллелизмом с помощью технологии параллельного доступа посредством многоверсионности (англ. MVCC [19]), которая предоставляет каждой транзакции «снимок» текущего состояния базы данных, позволяя вносить изменения, не затрагивая другие транзакции. Это

в значительной степени устраняет необходимость в блокировках чтения (англ. `read lock` [20]) и гарантирует, что база данных поддерживает принципы ACID.

## **Oracle Database**

Oracle Database [21] – объектно-реляционная система управления базами данных компании Oracle [22], являющаяся, на момент написания курсовой работы, самой популярной в мире [23].

Oracle Database предоставляет транзакции, соответствующие принципам ACID, поддержку триггеров, хранимых процедур и внешние ключи. Особенностью рассматриваемой СУБД является функциональность работы с большими массивами данных, реализованная с использованием таких методов параллелизма, как механизм блокировки, обеспечивающий монопольное использование таблицы одной транзакцией и метод временных меток, обеспечивающий сериализацию и планирование транзакций.

## **MySQL**

MySQL [24] – свободная реляционная система управления базами данных, разрабатываемая и поддерживаемая корпорацией Oracle.

Рассматриваемая СУБД имеет два основных движка хранения данных: InnoDB [25] и myISAM [26]. Движок InnoDB полностью совместим с принципами ACID, в отличие от движка myISAM. Параллелизм реализован с помощью механизма блокировок, который обеспечивает монопольный доступ к данным.

### **3.2.2 Обзор NoSQL СУБД**

#### **Datomic**

Datomic [27] – это распределённая база данных класса NoSQL, реализующая язык логических операций Datalog [28].

Особенностью рассматриваемой СУБД является связь каждой транзакции с текущим временем, база данных хранит историю всех операций, что позволяет получить её состояние в любой момент времени, такой подход позво-

ляет реализовать режим просмотра истории изменения данных, являющийся аналогом журналирования.

Datomic предоставляет транзакции, соответствующие принципам ACID и состоит из механизма изменения состояния базы данных – транзактора, сервиса хранилища данных и пользовательского сервиса, который отсылает данные на транзактор. Оптимальная скорость работы достигается за счёт зарезервированных инструкций и консистентности данных, встроенного в транзактор механизма кэширования запросов как на чтение, так и на запись, и отсутствия планировщика транзакций. Унификация данных структурой вида «ключ-значение» позволяет достичь согласованности, в качестве ключа может быть как зарезервированное слово, так и атрибут сущности или хранимая процедура.

Главной особенностью является возможность пользователя полностью влиять на системные настройки и отсутствие явных правил формирования запроса. Неумелое обращение с данной СУБД может привести к результатам на порядок хуже, чем в традиционных реализациях, например, отсутствие планировщика транзакций может значительно замедлить выполнение составного запроса.

## **Redis**

Redis [29] – резидентная система управления базами данных класса NoSQL с открытым исходным кодом, используемая как для реализации кэшей и брокеров сообщений, так и для хранения данных. Основная структура представлена отношением вида «ключ-значение».

Рассматриваемая СУБД хранит данные в оперативной памяти, снабжена механизмом «снимков» и журналирования, предоставляет операции для реализации механизма обмена сообщениями в шаблоне «издатель-подписчик», позволяющие пользователю отслеживать изменения в системе, предоставляет встроенный механизм репликации данных, основанный на принципе master-slave [30], поддерживает пакетную обработку команд и высокоуровневые опе-

рации над наборами данных.

Отличие Redis от других СУБД заключается в том, что значение, соответствующее определённому ключу, не ограничивается строковым типом. Поддерживаются следующие абстрактные типы данных:

- строки;
- списки;
- множества;
- хеш-таблицы;
- упорядоченные множества.

### **Вывод**

В качестве СУБД была выбрана NoSQL Datomic с сервисом хранения PostgreSQL. Данное решение подходит для обработки большого объёма данных, удовлетворяет требованию в горизонтальном масштабировании, транзакции соответствуют свойствам ACID, отсутствует строгая типизация данных, а встроенный механизм кэширования позволяет обеспечить оптимальное время ожидания. Сравнительная характеристика СУБД представлена в таблице 1.

Таблица 1 – Сравнение СУБД

БД	Строгая типизация	ACID	Масштабируемость	Кэш
PostgreSQL	+	+	+	-
Redis	-	-	±	-
MySQL	+	-	+	-
Oracle	+	+	+	-
Datomic	-	+	+	+

### **3.3 Листинги кода**

В данном подразделе будут приведены листинги кода, реализующие отдельно взятую функциональность приложения и базы данных. Будут рассмотрены: функции создания схемы и пример её построения, функции доступа к

данным, использующие обход графа по обратным ссылкам [13] с селектором, функция автоматического создания уникального кода и функция осуществления запроса к серверу приложения.

### **3.3.1 Создание схемы базы данных**

Для упрощения создания схемы был реализован генератор схемы, позволяющий автоматически создавать сущность с предустановленными атрибутами и добавлять новые, передавая их в коротком формате, не используя пространство имён. Генератор схемы и пример создания сущности отражён в приложении А.

Учитывая особенности выбранной СУБД, ключевым словом, определяющим атрибут, является ключ `:db/ident`, а значением может быть любое ключевое слово, например, `:namespace/type`. Тип задаётся ключевым словом `:db/valueType` и может принимать следующие значения:

- `:db.type/string` – строковый тип;
- `:db.type/boolean` – логический тип;
- `:db.type/long` – целочисленный тип;
- `:db.type/float` – тип с плавающей точкой;
- `:db.type/keyword` – тип ключевого слова;
- `:db.type/ref` – ссылка на пользовательский тип или сущность.

### **3.3.2 Доступ к атрибутам сущности базы данных**

Datomic предоставляет сущность в виде графа, где связь с другой сущностью реализована с помощью обратной ссылки. Для доступа к нужным атрибутам сущности был реализован механизм обхода графа по обратным ссылкам с использованием селектора атрибутов. Чтобы явно указать, какие поля сущности нужно получить, необходимо передать соответствующий селектор формата `{ns [[attr op]]}`, где `ns` – пространство имён сущности, `attr` – нужный атрибут, а `op` – операция, определяемая типом атрибута: `atomic` по умолчанию, `pull-ref` – обход графа по обратной ссылке. Реализация функции обхода графа сущности представлена в приложении Б.

### **3.4 Хранимые процедуры**

В качестве идентификатора сущности используется уникальный код, который не может одновременно соответствовать более, чем одной сущности. Для упрощения создания уникального кода средствами языка Clojure была реализована хранимая процедура с использованием системной сущности счётчика, вызываемая при осуществлении транзакции добавления, чтобы установить необходимое уникальное значение в соответствующий атрибут сущности. Реализация автоматического создания кода приведена в приложении В.

### **3.5 Взаимодействие с приложением**

Взаимодействие с приложением осуществляется посредством метода вызова удалённых процедур (анг. Remote Procedure Call, RPC [31]), состоящего в расширении механизма передачи управления и данных внутри программы, выполняющейся на одном узле, на передачу управления и данных через сеть. Средства удалённого вызова процедур предназначены для облегчения организации распределённых вычислений и создания распределённых клиент-серверных информационных систем. Наибольшая эффективность использования RPC достигается в тех приложениях, в которых существует интерактивная связь между удалёнными компонентами с небольшим временем ответов и относительно малым количеством передаваемых данных. Реализация удалённого вызова процедуры представлена в приложении Г.



## **4 Исследовательская часть**

В этом разделе приведены результаты исследования поведения инфраструктуры выбранной системы управления базами данных при высоких нагрузках, технические характеристики устройства, на котором проводилось исследование, и средства проведения исследования. По результатам проведённого исследования составлено заключение, включающее установленную зависимость производительности системы от объёма предоставляемой кэш-памяти.

### **4.1 Технические характеристики**

Исследование проводилось на ноутбуке при включённом режиме производительности и питании от сети. Во время исследования ноутбук был нагружен только системными процессами. Помещение и место проведения исследования предварительно были подготовлены во избежание перегрева технических элементов ноутбука. Технические характеристики устройства, на котором проводилось исследование:

- операционная система: Kali [32] Linux [33] 5.16.0-kali7-amd64;
- оперативная память: 8 Гб;
- размер SWAP-файла [34] кэша: 4 Гб;
- SSD-накопитель [35]: Samsung 980 NVMe M.2 512 Гб [36];
- процессор: Intel® Core™ i5-8250U [37] CPU @ 1.60GHz.

### **4.2 Тестирование отказоустойчивости**

Было проведено исследование в области отказоустойчивости и проверки настроек СУБД Datomic при высоких нагрузках на инфраструктуру транзактора. Исследование проводилось при следующих настройках: для кэширования запросов на транзакторе на SSD-накопителе был выделен SWAP-файл размером 4 Гб, количество одновременно использующих базу пользователей увеличивалось до 100 с интервалом в 30 секунд и шагом 1, каждый подключённый пользователь совершал непрерывные запросы на запись и чтение. Инструментом исследования был выбран кластер в системе Kubernetes [38], позволяющий

описать сценарий оркестрации приложений в едином конфигурационном файле и получать метрики в режиме реального времени. Результаты представлены на графиках 8 и 9.

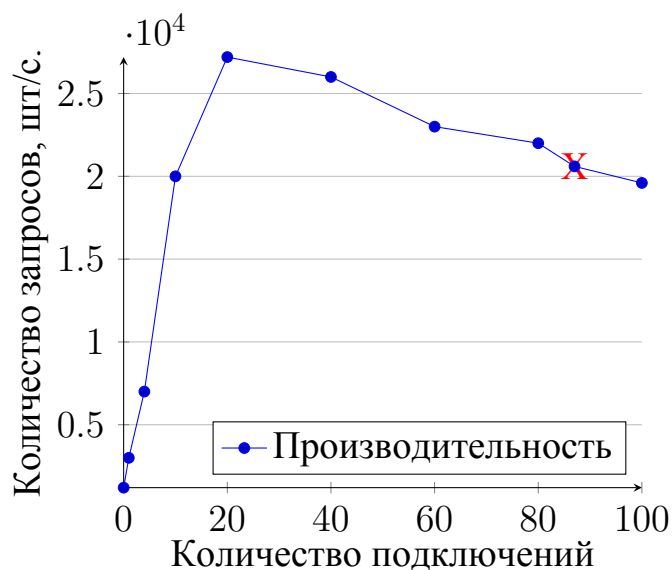


Рисунок 8 – Количество операций в секунду.

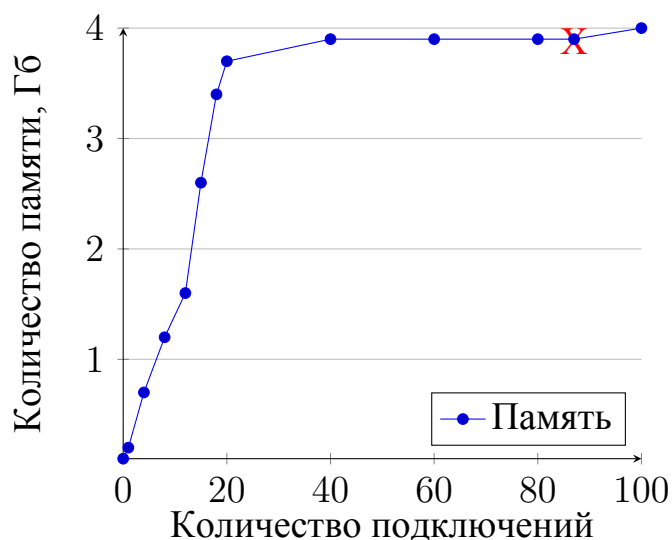


Рисунок 9 – Объем занимаемого кэша.

На основе полученных данных установлена зависимость производительности системы от размера занятого кэша, при его переполнении система снижает свою производительность до 17% в пике нагрузки. Это связано с особенностью кэширования: чем больше уникальных запросов, тем больше объема

занимает кэш. В ходе исследования вызвать отказ в обслуживании системы не удалось, устройство, на котором проводилось исследование, вышло из строя намного раньше инфраструктуры системы – примерно на 87 подключённом пользователе вышел из строя SSD-накопитель, на котором был размещён кэш для транзактора, что привело к досрочному завершению исследования. Данные на графиках были экстраполированы для отображения недополученных результатов.

## **Заключение**

В рамках выполнения курсового проекта были выполнены следующие задачи:

- проведён анализ предметной области и готовых решений в области баз данных и систем управления базами данных;
- спроектирована база данных, описаны её компоненты и связи;
- реализованы база данных и программное обеспечение для работы с ней;
- проведено исследование устойчивости реализованной системы к высоким нагрузкам.

Поставленная цель достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Положение о порядке разработки и утверждение рабочей программы дисциплины [Электронный ресурс]. Режим доступа: [https://www.volgmed.ru/uploads/files/2010-11/1180-polozhenie\\_o\\_poryadke\\_razrabotki\\_i\\_utverzhdeniya\\_rabochej\\_programmy\\_uchebnoj\\_discipliny\\_\(kurs\).doc](https://www.volgmed.ru/uploads/files/2010-11/1180-polozhenie_o_poryadke_razrabotki_i_utverzhdeniya_rabochej_programmy_uchebnoj_discipliny_(kurs).doc) (дата обращения: 10 июня 2022 г.).
2. Положение об оценочных материалах (фондах оценочных средств) [Электронный ресурс]. Режим доступа: <http://www.osu.ru/doc/3825> (дата обращения: 10 июня 2022 г.).
3. Learning Management System (LMS) — HSE [Электронный ресурс]. Режим доступа: [https://www.hse.ru/en/studyspravka/lms\\_student/](https://www.hse.ru/en/studyspravka/lms_student/) (дата обращения: 10 июня 2022 г.).
4. Microsoft Word - Word Processing Software | Microsoft 365 [Электронный ресурс]. Режим доступа: <https://www.microsoft.com/en-us/microsoft-365/word> (дата обращения: 10 июня 2022 г.).
5. Образовательные стандарты | МГТУ им. Н. Э. Баумана [Электронный ресурс]. Режим доступа: <https://bmstu.ru/plain/eduStandarts/> (дата обращения: 10 июня 2022 г.).
6. МГТУ им. Н. Э. Баумана – Официальный сайт [Электронный ресурс]. Режим доступа: <https://bmstu.ru/> (дата обращения: 10 июня 2022 г.).
7. Что такое база данных | Oracle Россия и СНГ [Электронный ресурс]. Режим доступа: <https://www.oracle.com/ru/database/what-is-database/> (дата обращения: 10 июня 2022 г.).

8. Что такое СУБД - RU-CENTER [Электронный ресурс]. Режим доступа: [https://www.nic.ru/help/chto-takoe-subd\\_8580.html](https://www.nic.ru/help/chto-takoe-subd_8580.html) (дата обращения: 10 июня 2022 г.).
9. What is OLTP? | IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/cloud/learn/oltp> (дата обращения: 10 июня 2022 г.).
10. What is OLAP? | IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/cloud/learn/olap> (дата обращения: 10 июня 2022 г.).
11. Что такое NoSQL? | Amazon AWS [Электронный ресурс]. Режим доступа: <https://aws.amazon.com/ru/nosql/> (дата обращения: 10 июня 2022 г.).
12. Chen Peter Pin-Shan. The Entity-Relationship Model—toward a Unified View of Data // ACM Trans. Database Syst. New York, NY, USA, 1976. mar. Т. 1, № 1. с. 9–36.
13. Linking Relationships with Backref [Электронный ресурс]. Режим доступа: <https://docs.sqlalchemy.org/en/14/orm/backref.html> (дата обращения: 10 июня 2022 г.).
14. Clojure [Электронный ресурс]. Режим доступа: <https://clojure.org/> (дата обращения: 10 июня 2022 г.).
15. ClojureScript [Электронный ресурс]. Режим доступа: <https://clojurescript.org/> (дата обращения: 10 июня 2022 г.).
16. PostgreSQL: Документация. [Электронный ресурс]. Режим доступа: <https://postgrespro.ru/docs/postgresql/> (дата обращения: 10 июня 2022 г.).
17. PostgreSQL: вчера, сегодня, завтра [Электронный ресурс]. Режим доступа: <https://postgrespro.ru/blog/media/17768> (дата обращения: 10 июня 2022 г.).

18. Транзакции, ACID, CAP | GeekBrains [Электронный ресурс]. Режим доступа: [https://gb.ru/posts/acid\\_cap\\_transactions](https://gb.ru/posts/acid_cap_transactions) (дата обращения: 10 июня 2022 г.).
19. Documentation: 12: 13.1. Introduction - PostgreSQL [Электронный ресурс]. Режим доступа: <https://www.postgresql.org/docs/12/mvcc-intro.html> (дата обращения: 10 июня 2022 г.).
20. Применение блокировок чтения/записи | IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/docs/ru/aix/7.2?topic=programming-using-readwrite-locks> (дата обращения: 10 июня 2022 г.).
21. SQL Language | Oracle[Электронный ресурс]. Режим доступа: <https://www.oracle.com/database/technologies/appdev/sql.html> (дата обращения: 10 июня 2022 г.).
22. Oracle | Integrated Cloud Applications and Platform Services [Электронный ресурс]. Режим доступа: <https://www.oracle.com/index.html> (дата обращения: 10 июня 2022 г.).
23. DB-Engines Ranking [Электронный ресурс]. Режим доступа: <https://db-engines.com/en/ranking> (дата обращения: 10 июня 2022 г.).
24. MySQL Database Service is a fully managed database service to deploy cloud-native applications. [Электронный ресурс]. Режим доступа: <https://www.mysql.com/> ((дата обращения: 10 июня 2022 г.).
25. MySQL Reference Manual 8.0: The InnoDB Storage Enginee [Электронный ресурс]. Режим доступа: <https://dev.mysql.com/doc/refman/8.0/en/innodb-storage-engine.html> (дата обращения: 10 июня 2022 г.).

26. MySQL Reference Manual 16.2: The MyISAM Storage Engine [Электронный ресурс]. Режим доступа: <https://dev.mysql.com/doc/refman/8.0/en/myisam-storage-engine.html> (дата обращения: 10 июня 2022 г.).
27. Overview | Datomic [Электронный ресурс]. Режим доступа: <https://docs.datomic.com/on-prem/overview/overview.html> (дата обращения: 10 июня 2022 г.).
28. Datalog: Deductive Database Programming [Электронный ресурс]. Режим доступа: <https://docs.racket-lang.org/datalog/> (дата обращения: 10 июня 2022 г.).
29. Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker [Электронный ресурс]. Режим доступа: <https://redis.io/> (дата обращения: 10 июня 2022 г.).
30. Tech Confronts Its Use of the Labels «Master» and «Slave» [Электронный ресурс]. Режим доступа: <https://www.wired.com/story/tech-confronts-use-labels-master-slave/> (дата обращения: 10 июня 2022 г.).
31. Remote Procedure Call – IBM Documentation [Электронный ресурс]. Режим доступа: <https://www.ibm.com/docs/en/aix/7.1?topic=concepts-remote-procedure-call> (дата обращения: 10 июня 2022 г.).
32. Our Most Advanced Penetration Testing Distribution, Ever. [Электронный ресурс]. Режим доступа: <https://kali.org/> (дата обращения: 10 июня 2022 г.).
33. Linux.org [Электронный ресурс]. Режим доступа: <https://www.linux.org/> (дата обращения: 10 июня 2022 г.).



34. SWAP File Definition – Tech definitions [Электронный ресурс]. Режим доступа: [https://techterms.com/definition/swap\\_file](https://techterms.com/definition/swap_file) (дата обращения: 10 июня 2022 г.).
35. SSD Definition – Tech definitions [Электронный ресурс]. Режим доступа: <https://techterms.com/definition/ssd> (дата обращения: 10 июня 2022 г.).
36. Samsung NVMe 980 M.2 512GB [Электронный ресурс]. Режим доступа: <https://www.samsung.com/ru/memory-storage/nvme-ssd/980-480gb-nvme-pcie-gen-3-mz-v8v500bw/> (дата обращения: 10 июня 2022 г.).
37. Intel Processors [Электронный ресурс]. Режим доступа: <https://www.intel.com/content/www/us/en/products/processors/core/i5-processors.html> (дата обращения: 10 июня 2022 г.).
38. Что такое Kubernetes | Kubernetes [Электронный ресурс]. Режим доступа: <https://kubernetes.io/ru/docs/concepts/overview/what-is-kubernetes/> (дата обращения: 10 июня 2022 г.).
39. Java Platform Overview [Электронный ресурс]. Режим доступа: <https://docs.oracle.com/javase/7/docs/technotes/guides/> (дата обращения: 10 июня 2022 г.).

## ПРИЛОЖЕНИЕ А

### Листинг 1: Создание схемы

```
1 are
```

## ПРИЛОЖЕНИЕ Б

### Листинг 2: Доступ к сущностям

```
1 you
```

## ПРИЛОЖЕНИЕ В

### Листинг 3: Автогенерация уникального кода

```
1 mad
```

## ПРИЛОЖЕНИЕ Г

### Листинг 4: Осуществление RPC запроса

```
l enough?
```