

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1 Аналитическая часть</b>	<b>4</b>
1.1 Формализация задачи . . . . .	4
1.2 Структура рабочей программы дисциплины . . . . .	5
1.3 Базы данных и системы управления базами данных . . . . .	8
1.4 Хранение данных о рабочих программах дисциплины . . . . .	8
1.4.1 Классификация баз данных по способу хранения . . . . .	9
1.4.2 Выбор модели хранения данных для решения задачи	10
1.4.3 Обзор СУБД с построчным хранением . . . . .	10
1.4.4 Выбор СУБД для решения задачи . . . . .	11
1.5 Кэширование данных . . . . .	12
1.5.1 Проблемы кэширования данных . . . . .	12
1.5.2 Обзор in-memory СУБД . . . . .	12
1.5.3 Выбор СУБД для решения задачи . . . . .	12
1.6 Формализация данных . . . . .	12
1.6.1 База данных рабочих программ дисциплин . . . . .	12
1.6.2 База данных кэшируемой информации . . . . .	12
<b>2 Конструкторская часть</b>	<b>14</b>
2.1 Проектирование отношений сущностей . . . . .	14
2.2 Проектирование базы данных рабочих программ дисциплин	14
2.3 Проектирование базы данных кэширования . . . . .	14
<b>3 Технологическая часть</b>	<b>15</b>
3.1 Архитектура приложения . . . . .	15
3.2 Средства реализации . . . . .	15
3.3 Детали реализации . . . . .	15
3.4 ? REST ? . . . . .	15
<b>4 Исследовательская часть</b>	<b>16</b>
4.1 Постановка эксперимента . . . . .	16
4.1.1 Цель эксперимента . . . . .	16

4.1.2	Описание эксперимента . . . . .	16
4.1.3	Результат эксперимента . . . . .	16
	<b>Заключение</b>	<b>17</b>
	<b>Литература</b>	<b>18</b>

# Введение

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

Хранение, обработка и анализ информации, находящейся в рабочей программы дисциплины может пригодиться для различных систем, например, системы управления обучения (англ. Learning Management System (LMS) [2]). Такой интерфейс может предоставить пользователю системы (в данном случае преподавателю) получать и редактировать информацию о рабочей программы дисциплины в режиме онлайн, например, в личном кабинете пользователя.

Рабочая программа дисциплины обычно представлена в виде документа в формате Microsoft Word [3], что накладывает ограничения на автоматизированную программную обработку и анализ информации, предоставленной в рабочей программы дисциплины.

Цель работы – реализовать программное обеспечение для хранения, редактирования и удаления данных о рабочих программах дисциплин.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- проанализировать варианты представления данных и выбрать подходящий вариант для решения задачи;
- проанализировать системы управления базами данных и выбрать подходящую систему для хранения данных;
- спроектировать базу данных, описать ее сущности и связи;
- реализовать интерфейс для доступа к базе данных;
- реализовать программное обеспечение, которое позволит получить доступ к данным по средствам REST API [4].

# 1 Аналитическая часть

В данном разделе описана структура рабочей программы дисциплины. Представлен анализ способов хранения данных и систем управления базами данных, оптимальных для решения поставленной задачи. Описаны проблемы кэшированных данных и представлены методы их решения.

## 1.1 Формализация задачи

Каждая дисциплина, преподаваемая в высшем учебном заведении, имеет свою рабочую программу. В ней хранятся различная информация о дисциплине: стандарт, содержание, объем, 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: 3+, 3++ и другие [5]. Образовательный стандарт – это совокупность обязательных требований к образованию определенного уровня и (или) к профессии, специальности. Каждый образовательный стандарт для каждого направления подготовки обучаемого имеет свою компетенцию. Компетенция – некоторый свод информации, о том что должен знать, уметь и какими навыками должен обладать выпускник, успешно осовевший дисциплину.

Каждому направлению подготовки, но одной и той же рабочей программы дисциплины сопоставлены различные компетенции (которые, в свою очередь, различны в каждом образовательном стандарте). Например, есть два студента, успешно осовевшие дисциплину «Физика». Один из них обучается по направлению подготовки «Экономика», а второй по направлению «Теплофизика и теоретическая теплотехника». Очевидно, что второй студент должен владеть большими знаниями о данной дисциплине.

Коды компетенций для каждого образовательного стандарта отличаются. К сожалению, не редки случаи, когда при переходе на новую образовательную программу меняется только код и название компетенции – содержание компетенции остается абсолютно точно таким же. В связи с

этим, возникает возможность автоматизации перевода документов рабочей программы дисциплины на новый стандарт.

Кроме того, имея данные о всех рабочих программах дисциплин, можно адаптировать какие-либо из них под выбранные направления подготовки. Например, рассчитать оптимальную нагрузку по данной дисциплине для студентов обучающихся на данном направлении.

Далее, в качестве примера, будем рассматривать рабочую программу дисциплины «Информатика», соответствующую образовательному стандарту 3++, разработанную и преподаваемую в МГТУ им. Н. Э. Баумана [6].

## **1.2 Структура рабочей программы дисциплины**

Структура файлов рабочих программ дисциплин можно разниться от ВУЗа к ВУЗу, но, внутри одного ВУЗа, скорее всего, все программы имеют одну и ту же (или схожую) структуру. Рабочая программа дисциплины «Информатика» имеет следующие разделы:

1. титульный лист;
2. планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы;
3. место дисциплины в структуре образовательной программы;
4. объем 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. 1.1). Остальные разделы представлены в виде текстовой информации (рис. 1.2).

### 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	Объём по семестрам, ч		
	Всего	Семестры	
		1	2
Объём 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>

Рис. 1.1: Изображение таблицы с информацией о объеме дисциплины.

Интерес представляют разделы №1 (титульный лист), №2 (результаты обучения), №4 (объем дисциплины), №5 (содержание дисциплины) и №8 (перечень литературы).

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 нанoeлектроника», 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 «Нанoeинженерия».

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 нанoeлектроника», 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 «Нанoeинженерия».

Рис. 1.2: Изображение текстовой информации о месте дисциплины в структуре образовательной программы.

Раздел №2 содержит коды и описания компетенций для каждого направления подготовки. С помощью информации, полученной в этом разделе, можно попробовать автоматизировать перенос файла рабочей программы дисциплины с одной образовательной программы на другую.

В разделах №4 и №5 хранится информация о нагрузке и структуре дисциплины – эта информация может пригодиться для анализа нагрузки на студентов по выбранному направлению подготовки и структуризации рассматриваемой рабочей программы дисциплины.

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

В задаче разбора и хранения информации рабочей программы дисциплины важную роль имеет выбор модели хранения данных. Для персистентного хранения данных используются базы данных [7]. Для управления этими базами данных используется системы управления данными – СУБД [8]. Система управления базами данных – это совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

## **1.4 Хранение данных о рабочих программах дисциплины**

Система, разрабатываемая в рамках курсового проекта, предполагает собой приложение, которое является микросервисом [9] одной большой системы – системы управления обучения.

Предполагается, что доступ к разрабатываемому приложению будем иметь лишь только «ядро» этой системы. При этом, только у одного типа пользователя системы есть доступ к данным, хранящимся в приложении – преподавателю. Состояние гонки (англ. Race condition [10]) можно исключить - каждый преподаватель работает только с информацией из файлов, которые он самостоятельно загрузил в базу данных.

Для хранения данных о рабочей программы дисциплины необходимо использовать строго структурированную и типизированную базу данных, потому что вся информация, предоставленная в файлах программы имеет чётко выраженную структуру, которая не будет меняться от дисциплины к дисциплине.



### **1.4.1 Классификация баз данных по способу хранения**

Базы данных, по способу хранения, делятся на две группы – строковые и колоночные. Каждый из этих типов служит для выполнения для определенного рода задач.

#### **Строковые базы данных**

Строковыми базами данных называются такие базы данных, записи которых в памяти представляются построчно. Строковые базы данных используются в транзакционных системах (англ. OLTP [11]). Для таких систем характерно большое количество коротких транзакций с операциями вставки, обновления и удаления данных - INSERT, UPDATE, DELETE.

Основной упор в системах OLTP делается на очень быструю обработку запросов, поддержание целостности данных в средах с множественным доступом и эффективность, которая измеряется количеством транзакций в секунду.

Схемой, используемой для хранения транзакционных баз данных, является модель сущностей, которая включает в себя запросы, обращающиеся к отдельным записям. Так же, в OLTP-системах есть подробные и текущие данные.

#### **Колоночные базы данных**

Колоночными базами данных называются базы данных, записи которых в памяти представляются по столбцам. Колоночные базы данных используются в аналитических системах (англ. OLAP [12]). OLAP характеризуется низким объемом транзакций, а запросы часто сложны и включают в себя агрегацию. Время отклика для таких систем является мерой эффективности.

OLAP-системы широко используются методами интеллектуального анализа данных. В таких базах есть агрегированные, исторические данные, хранящиеся в многомерных схемах.

## 1.4.2 Выбор модели хранения данных для решения задачи

Для решения задачи построчное хранение данных преобладает над колоночным хранением по нескольким причинам:

- задача предполагает постоянное добавление и изменение данных;
- задача предполагает быструю отзывчивость на запросы пользователя;
- задача не предполагает выполнения аналитических запросов;

## 1.4.3 Обзор СУБД с построчным хранением

В данном подразделе будут рассмотрены популярные построчные СУБД, которые могут быть использованы для реализации хранения в разрабатываемом программном продукте.

### PostgreSQL

PostgreSQL [13] – это свободно распространяемая объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных [14].

PostgreSQL предоставляет транзакции со свойствами атомарности, согласованности, изоляции, долговечности (ACID [15]), 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 управления многоверсионным параллелизмом (англ. MVCC [16]). Эта технология дает 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 [17]) и гарантирует, что база данных поддерживает принципы ACID.

## **Oracle**

Oracle [18] –

## **MySQL**

MySQL [19] –

### **1.4.4 Выбор СУБД для решения задачи**

Для решения задачи была выбрана СУБД PostgreSQL, потому что данная СУБД имеет поддержку языка `plpython3u` [20], который упрощает процесс интеграции базы данных в разрабатываемое приложение. Кроме того, PostgreSQL проста в развертывании.

## 1.5 Кэширование данных

### 1.5.1 Проблемы кэширования данных

### 1.5.2 Обзор in-memory СУБД

### 1.5.3 Выбор СУБД для решения задачи

## 1.6 Формализация данных

### 1.6.1 База данных рабочих программ дисциплин

### 1.6.2 База данных кэшируемой информации

База данных кэшируемых значений должна хранить значения без дополнительной обработки. Данные должны быть актуальны и синхронизированы с основным хранилищем: кэш должен обновляться после каждой транзакции. Кроме того, нужно ограничить размер кэша и добавить вытеснение из него, например, с помощью политики вытеснения LRU [21] (Last Recently Used).

## Вывод

В данном разделе:

- рассмотрена структура рабочей программы дисциплины и выявлены её наиболее интересные части;
- проанализированы способы хранения информации для система и выбраны оптимальные способы для решения поставленной задачи;

- проведен анализ СУБД, используемых для решения задачи и также выбраны оптимальные информационные системы;
- рассмотрена проблема актуальности кэшируемых данных и предложено ее решение;
- формализованы данные, используемые в системе.

## 2 Конструкторская часть

2.1 Проектирование отношений сущностей

2.2 Проектирование базы данных рабочих программ дисциплин

2.3 Проектирование базы данных кэширования

Вывод

## 3 Технологическая часть

### 3.1 Архитектура приложения

### 3.2 Средства реализации

### 3.3 Детали реализации

### 3.4 ? REST ?

### Вывод

## 4 Исследовательская часть

### 4.1 Постановка эксперимента

#### 4.1.1 Цель эксперимента

#### 4.1.2 Описание эксперимента

#### 4.1.3 Результат эксперимента

### Вывод



# Заключение

# Литература

- [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\\_\(kursa\).doc](https://www.volgmed.ru/uploads/files/2010-11/1180-polozhenie_o_poryadke_razrabotki_i_utverzhdeniya_rabochej_programmy_uchebnoj_discipliny_(kursa).doc) (дата обращения: 07.06.2021).
- [2] 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: 07.06.2021).
- [3] Microsoft Word - Word Processing Software | Microsoft 365 [Электронный ресурс]. Режим доступа: <https://www.microsoft.com/en-us/microsoft-365/word> (дата обращения: 07.06.2021).
- [4] What is a REST API? - Red Hat [Электронный ресурс]. Режим доступа: <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (дата обращения: 07.06.2021).
- [5] Образовательные стандарты | МГТУ им. Н. Э. Баумана [Электронный ресурс]. Режим доступа: <https://bmstu.ru/plain/eduStandarts/> (дата обращения: 07.06.2021).
- [6] МГТУ им. Н. Э. Баумана – Официальный сайт [Электронный ресурс]. Режим доступа: <https://bmstu.ru/> (дата обращения: 07.06.2021).
- [7] Что такое база данных | Oracle Россия и СНГ [Электронный ресурс]. Режим доступа: <https://www.oracle.com/ru/database/what-is-database/> (дата обращения: 07.06.2021).
- [8] Что такое СУБД - RU-CENTER [Электронный ресурс]. Режим доступа: [https://www.nic.ru/help/что-такое-sbd\\_8580.html](https://www.nic.ru/help/что-такое-sbd_8580.html) (дата обращения: 07.06.2021).
- [9] Что такое микросервисная архитектура: простое объяснение | MCS Mail.ru [Электронный ресурс]. Режим доступа: <https://mcs.mail.ru/blog/prostym-jazykom-o-mikroservisnoj-arhitekture> (дата обращения: 07.06.2021).

- [10] Race conditions and deadlocks - Microsoft Docs [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/en-us/troubleshoot/dotnet/visual-basic/race-conditions-deadlocks> (дата обращения: 07.06.2021).
- [11] What is OLTP? | IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/cloud/learn/oltp> (дата обращения: 07.06.2021).
- [12] What is OLAP? | IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/cloud/learn/olap> (дата обращения: 07.06.2021).
- [13] PostgreSQL: Документация. [Электронный ресурс]. Режим доступа: <https://postgrespro.ru/docs/postgresql/> (дата обращения: 07.06.2021).
- [14] PostgreSQL: вчера, сегодня, завтра [Электронный ресурс]. Режим доступа: <https://postgrespro.ru/blog/media/17768> (дата обращения: 07.06.2021).
- [15] Транзакции, ACID, CAP | GeekBrains [Электронный ресурс]. Режим доступа: [https://gb.ru/posts/acid\\_cap\\_transactions](https://gb.ru/posts/acid_cap_transactions) (дата обращения: 07.06.2021).
- [16] Documentation: 12: 13.1. Introduction - PostgreSQL [Электронный ресурс]. Режим доступа: <https://www.postgresql.org/docs/12/mvcc-intro.html> (дата обращения: 07.06.2021).
- [17] 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: 07.06.2021).
- [18] SQL Language | Oracle [Электронный ресурс]. Режим доступа: <https://www.oracle.com/database/technologies/appdev/sql.html> (дата обращения: 07.06.2021).
- [19] MySQL Database Service is a fully managed database service to deploy cloud-native applications. [Электронный ресурс]. Режим доступа: <https://www.mysql.com/> (дата обращения: 07.06.2021).

- [20] PostgreSQL: Документация: 9.6: 44.1. Python 2 и Python 3. [Электронный ресурс]. Режим доступа: <https://postgrespro.ru/docs/postgresql/9.6/plpython-python23> (дата обращения: 07.06.2021).
- [21] LRU, метод вытеснения из кэша | Habr [Электронный ресурс]. Режим доступа: <https://habr.com/ru/post/136758/> (дата обращения: 07.06.2021).