



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 базы данных для идентификации
пользователей при осуществлении
контрольно-пропускного режима с использованием
документов, удостоверяющих личность»*

Студент ИУ7-61Б
(Группа)

(Подпись, дата)

Постнов С. А.
(И. О. Фамилия)

Руководитель курсовой работы

(Подпись, дата)

Кудрявцев М. А.
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитический раздел	4
1.1 Формализация задачи и данных	4
1.2 Описание предметной области	5
1.3 Реляционные базы данных	6
1.3.1 Oracle Database	6
1.3.2 MySQL	7
1.3.3 Microsoft SQL Server	7
1.3.4 PostgreSQL	8
1.4 Нереляционные базы данных	9
1.4.1 Redis	10
1.4.2 MongoDB	10
1.5 Вывод	11
2 Конструкторский раздел	12
2.1 Диаграмма вариантов использования	12
2.2 Диаграмма базы данных	12
2.3 Вывод	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

ВВЕДЕНИЕ

В настоящее время все больше компаний вводят систему контрольно-пропускного режима, что объясняется некоторыми основными причинами [1]:

- 1) контроль времени прихода и ухода работника;
- 2) обеспечение только санкционированного доступа в офис (здание, помещение);
- 3) обеспечение сохранности имущества работодателя.

Целью курсовой работы является разработка базы данных для идентификации пользователей при осуществлении контрольно-пропускного режима с использованием документов, удостоверяющих личность.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) формализовать задачу и данные;
- 2) провести анализ существующих СУБД;
- 3) спроектировать диаграмму вариантов использования, диаграмму базы данных и ограничения целостности;
- 4) выбрать средства реализации базы данных и приложения (в том числе выбор СУБД);
- 5) разработать базу данных и приложение;
- 6) описать интерфейс доступа к базе данных;
- 7) провести исследование зависимости скорости получения и сохранения изображения от типа СУБД.

1 Аналитический раздел

В аналитическом разделе будут формализованы задача и данные, рассмотрены существующие СУБД и составлены сравнительные таблицы.

1.1 Формализация задачи и данных

В соответствии с техническим заданием необходимо разработать приложение для идентификации пользователей при осуществлении контрольно-пропускного режима с использованием документов, удостоверяющих личность. Необходимо реализовать промежуточный слой бизнес-логики для взаимодействия с базой данных, а также графический интерфейс, обеспечивающий возможность авторизации и регистрации пользователя, поиска информации о человеке (людях) по следующим заданным параметрам:

- 1) номер документа, удостоверяющего личность;
- 2) фамилия;
- 3) номер КПП.

По результатам поиска должна предоставляться возможность просмотреть фотографию любого найденного человека с целью дальнейшей идентификации с минимальным временем ожидания.

Для создания базы данных для идентификации пользователей необходимо выделить следующие сущности:

- 1) компания;
- 2) сотрудник службы безопасности;
- 3) сотрудник компании;
- 4) информационная карточка;
- 5) документ, удостоверяющий личность;
- 6) фото;
- 7) поле документа;

- 8) контрольно-пропускной пункт (КПП);
- 9) проход.

На рисунке 1.1 представлена ER - диаграмма сущностей базы данных в нотации Чена.

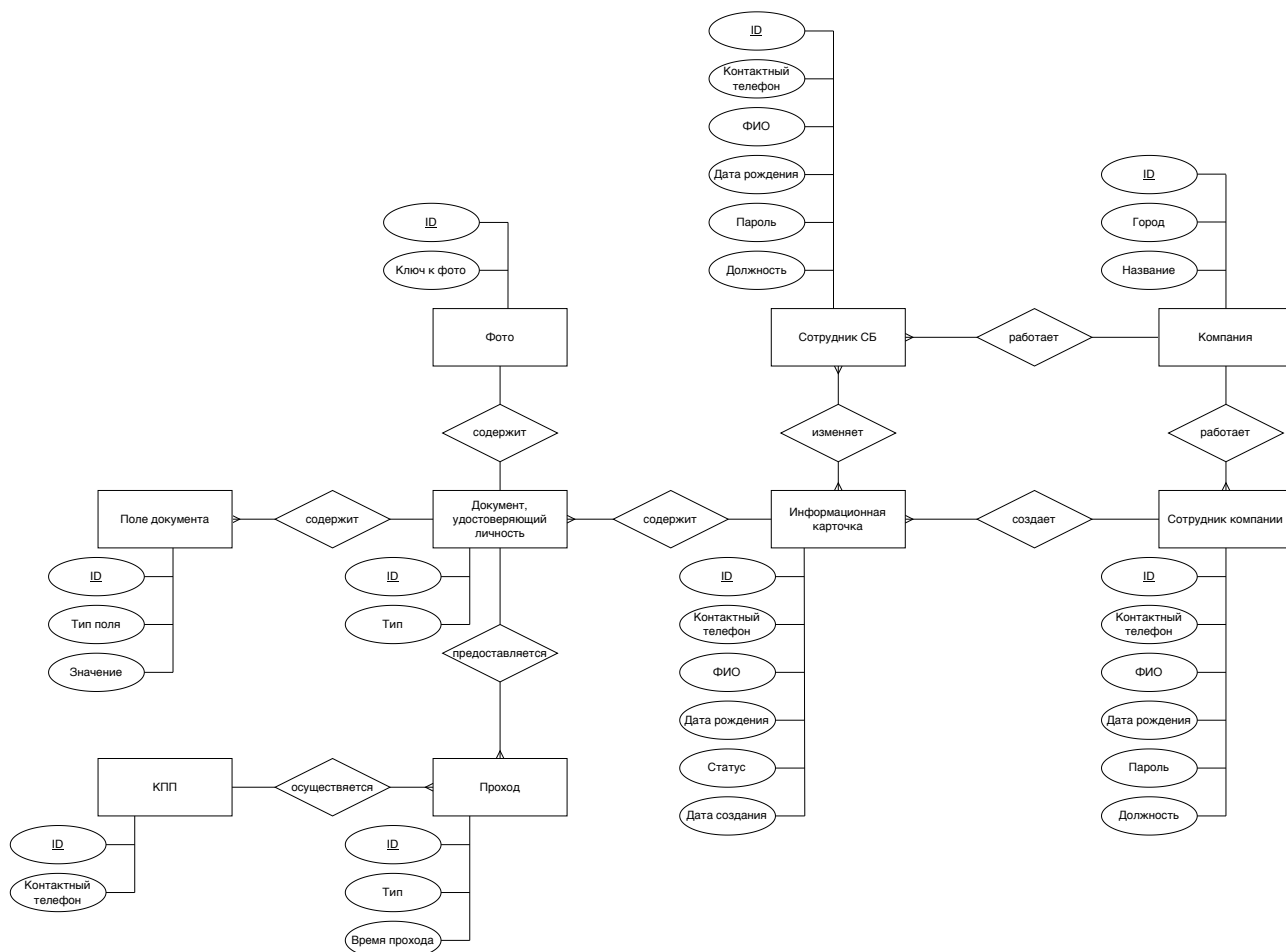


Рисунок 1.1 – ER - диаграмма сущностей базы данных

1.2 Описание предметной области

Информационная система – совокупность определенным образом организованных данных (база данных – БД) и комплекса аппаратно-программных средств хранения и манипулирования данными (система управления – СУ). База данных – основа информационной системы, выступающая в роли хранилища данных, которой необходимо управлять при помощи СУБД [2].

Исходя из потребности в манипуляции более сложными объектами (изображениями), а не только набором некоторых простых символьных или

текстовых данных, необходимо рассмотреть классификацию СУБД по модели данных [3]:

- 1) реляционные;
- 2) нереляционные («NoSQL»).

1.3 Реляционные базы данных

Основоположником теории реляционных баз данных является британский ученый Эдгар Кодд, который в 1970 году опубликовал первую работу по реляционной модели данных. Реляционная база данных – это множество взаимосвязанных именованных отношений («таблиц»). Отношение – это информационная модель реального объекта («сущности») предметной области, формально представленная множеством однотипных кортежей. Кортеж отношения представляет экземпляр моделируемого объекта, свойства которого определяются значениями соответствующих атрибутов («полей») кортежа [2].

По мере значительного повышения мощности компьютеров и соединения их в сети цели их использования стали все более разнообразными. Применение реляционных БД распространилось на широкое множество сценариев далеко за пределами первоначальной обработки коммерческих данных. Работа большей части Интернета до сих пор обеспечивается реляционными БД: онлайн-публикации, дискуссионные форумы, социальные сети, интернет-магазины, игры, предоставляемые как сервис, офисные приложения и многое другое [3].

Далее будут рассмотрены самые популярные реляционные баз данных [4]:

- 1) Oracle Database;
- 2) MySQL;
- 3) Microsoft SQL Server;
- 4) PostgreSQL.

1.3.1 Oracle Database

Oracle Database – флагманский продукт компании Oracle. Это популярная система управления базами данных и хранения данных,

используемая организациями по всему миру для управления своими данными и их хранения. Организации используют Oracle Database в различных отраслях, включая обслуживание конфиденциальных хранилищ данных и отслеживание крупных финансовых транзакций в режиме реального времени.

Основными преимуществами СУБД Oracle Database являются [5]:

- 1) многоверсионность данных для управления параллельными транзакциями;
- 2) поддержка ACID;
- 3) автоматический мониторинг и диагностика баз для выявления проблем производительности и возможность автоматической корректировки.

1.3.2 MySQL

MySQL – самая популярная в мире база данных с открытым исходным кодом. MySQL поддерживает многие наиболее популярные приложения, включая Netflix, Uber, Airbnb, Shopify и др. Поскольку MySQL имеет открытый исходный код, она включает в себя множество функций, разработанных в тесном сотрудничестве с пользователями на протяжении более чем 25 лет. Таким образом, большая часть языков программирования поддерживаются базой данных MySQL [6].

Основными преимуществами СУБД MySQL являются [7–9]:

- 1) открытый исходный код;
- 2) низкий порог входа;
- 3) удобство использования в случае простых приложений и запросов.

1.3.3 Microsoft SQL Server

Microsoft SQL Server – это реляционная система управления базами данных. Приложения и средства подключаются к экземпляру SQL Server или базе данных и взаимодействуют с помощью Transact-SQL. Среда MS SQL Server предоставляет множество различных функций для создания безопасных приложений баз данных. Платформа безопасности MS SQL Server управляет

доступом к защищаемым сущностям при помощи проверки подлинности и авторизации [10; 11].

Основными преимуществами СУБД Microsoft SQL Server являются [5]:

- 1) встроенный механизм репликации;
- 2) улучшенная производительность за счёт заранее скомпилированных модулей Transact-SQL;
- 3) поддержка всех существующих драйверов и фреймворков.

1.3.4 PostgreSQL

PostgreSQL – это мощная объектно-реляционная система баз данных с открытым исходным кодом, которая использует и расширяет язык SQL в сочетании со множеством функций, позволяющих безопасно хранить и масштабировать самые сложные рабочие нагрузки с данными. Истоки PostgreSQL восходят к 1986 году как части проекта Калифорнийского университета в Беркли, и более 35 лет ведется активная разработка на базовой платформе [12].

Основными преимуществами СУБД PostgreSQL являются [7—9]:

- 1) открытый исходный код и максимальное соответствие со стандартами SQL;
- 2) большое количество сторонних инструментов и библиотек;
- 3) соответствие требованиям ACID;
- 4) стабильность работы в условиях большого количества записей и сложных запросов.

Вывод

В таблице 1.1 приведены результаты сравнения рассмотренных реляционных СУБД, исходя из выдвинутых критериев [13].

Таблица 1.1 – Сравнительная таблица для рассмотренных реляционных СУБД

	PostgreSQL	Oracle Database	MySQL	Microsoft SQL Server
Лицензия	Открытая	Коммерческая	Открытая	Коммерческая
Поддержка ACID	Полная	Полная	Частичная	Полная
Возможность хранения бинарных данных	Есть	Есть	Есть	Есть

1.4 Нереляционные базы данных

Термин «NoSQL» был впервые применен в 1998 году Карло Строцци в качестве названия для его небольшой реляционной СУБД, которая не использовала язык SQL для манипулирования данными. С 2009 года термин «NoSQL» стал использоваться уже для обозначения растущего числа распределенных систем управления данными, которые отказывались от поддержки ACID-транзакций – одного из ключевых принципов работы с реляционными базами данных [14].

Существует несколько основных причин широкого внедрения NoSQL баз данных [3]:

- 1) потребность в больших возможностях масштабирования, чем у реляционных БД, включая обработку очень больших наборов данных или очень большую пропускную способность по записи;
- 2) специализированные запросные операции, плохо поддерживаемые реляционной моделью;
- 3) стремление к более динамичным и выразительным моделям данных.

Обычно NoSQL-системы делят, исходя из модели данных, на следующие основные классы [14]:

- 1) «ключ-значение»;
- 2) документные;
- 3) системы типа Google BigTable.

Необходимо рассмотреть только самые популярные нереляционные СУБД «ключ-значение» и документных СУБД соответственно, так как в рамках курсового проекта отсутствует потребность в работе с колоночными СУБД [4]:

- 1) Redis;
- 2) MongoDB.

1.4.1 Redis

Redis – система управления данными типа «ключ-значение» с открытым исходным кодом, написанная на С и поддерживающая достаточно богатую модель данных. Значения могут содержать не только строки, но и множества, списки и другие структуры данных. Redis работает в оперативной памяти, за счёт чего достигается высокая производительность. Клиентские библиотеки для работы с Redis доступны для большинства языков программирования, а сама система используется в таких крупных проектах, как Github, StackOverflow и др.

Основными преимуществами СУБД Redis являются [15]:

- 1) открытый исходный код;
- 2) быстрые чтение и запись;
- 3) гибкое использование как кеш-систему или брокер сообщений.

1.4.2 MongoDB

MongoDB – документная СУБД с открытым исходным кодом, написанная на С++ и разрабатываемая компанией 10gen. MongoDB обладает достаточно богатой функциональностью и является одной из самых популярных NoSQL-систем на данный момент и позволяет оперировать JSON-документами (хранимыми и передаваемыми в виде BSON – более компактного

двоичного представления JSON), объединяемыми в коллекции, которые, объединяются в базы данных. Заранее predeterminedенной схемы данных нет, то есть документы в одной коллекции могут содержать разные наборы полей [14].

Основными преимуществами СУБД MongoDB являются [15]:

- 1) открытый исходный код;
- 2) богатый язык запросов с возможностью гибкой выборки;
- 3) функциональные и простые в создании индексы;
- 4) гибкая сегментация и резервное копирование;
- 5) поддержка хранения как на диске, так и в оперативной памяти;
- 6) соответствие требованиям ACID.

Вывод

В таблице 1.2 приведены результаты сравнения рассмотренных нереляционных СУБД, исходя из выдвинутых критериев [15].

Таблица 1.2 – Сравнительная таблица для рассмотренных нереляционных СУБД

	MongoDB	Redis
Лицензия	Открытая	Открытая
Поддержка ACID	Полная	Отсутствует
Поддержка сложных запросов и расширенной фильтрации	Есть	Частичная
Модель хранения	Как на диске, так и в оперативной памяти	Только в оперативной памяти
Возможность хранения бинарных данных	Есть	Есть

1.5 Вывод

В аналитическом разделе были рассмотрены основные реляционные и нереляционные СУБД и составлены сравнительные таблицы.

2 Конструкторский раздел

В конструкторском разделе будут спроектированы диаграмма вариантов использования, диаграмма базы данных и ограничения целостности.

2.1 Диаграмма вариантов использования

На рисунке 2.1 представлена спроектированная диаграмма вариантов использования.

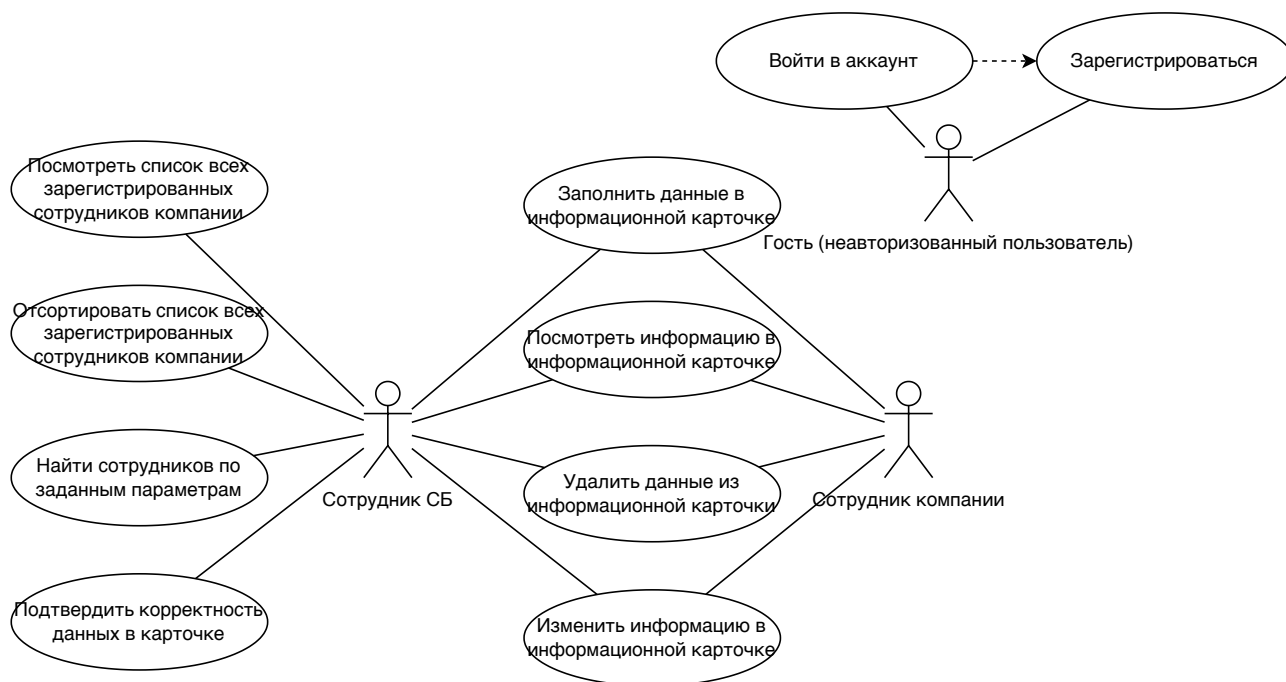


Рисунок 2.1 – Диаграмма вариантов использования

Исходя из спроектированной диаграммы вариантов использования 2.1, выделяются следующие роли:

- 1) Гость – неавторизованный сотрудник компании;
- 2) Сотрудник компании – пользователь, зарегистрированный в приложении;
- 3) Сотрудник службы безопасности (СБ) – сотрудник компании, обеспечивающий безопасность в рамках компании.

2.2 Диаграмма базы данных

На рисунке 2.2 представлена спроектированная диаграмма базы данных и ограничения целостности.

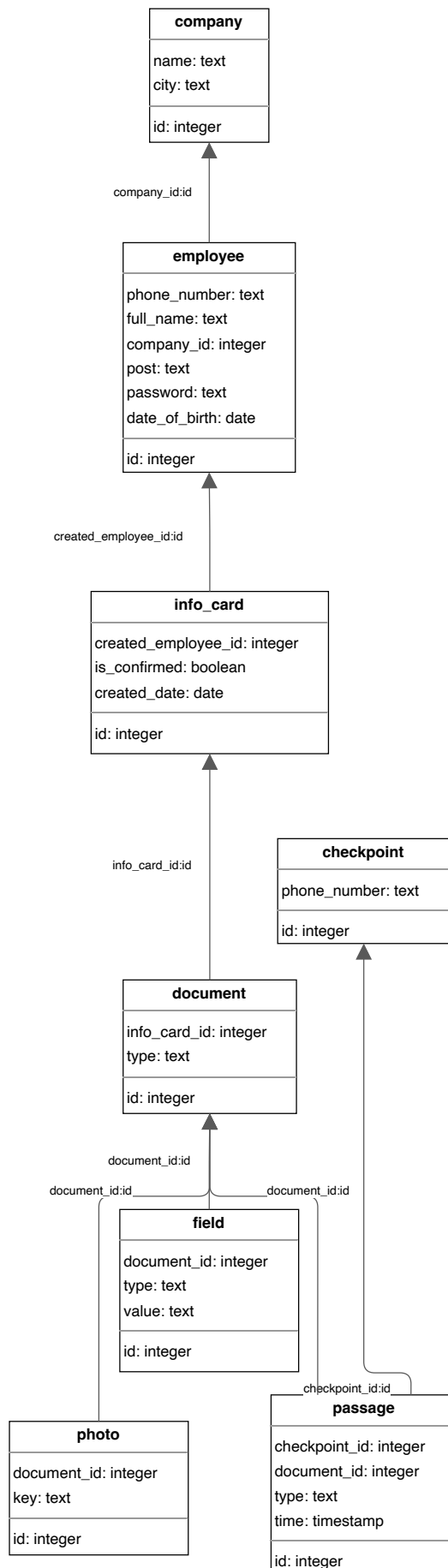


Рисунок 2.2 – Диаграмма базы данных и ограничения целостности

Для проектируемой базы данных необходимы следующие таблицы:

- 1) таблица компаний – «**company**»;
- 2) таблица сотрудников компании – «**employee**»;
- 3) таблица информационных карточек о сотруднике – «**info_card**»;
- 4) таблица документов, удостоверяющих личность – «**document**»;
- 5) таблица полей документов, удостоверяющих личность – «**field**»;
- 6) таблица фотографий сотрудников из документов, удостоверяющих личность – «**photo**»;
- 7) таблица контрольно-пропускных пунктов (КПП) – «**checkpoint**»;
- 8) связующая таблица проходов сотрудников через КПП – «**passage**».

2.3 Вывод

В конструкторском разделе были спроектированы диаграмма вариантов использования, диаграмма базы данных и ограничения целостности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Статья: Электронный пропускной режим: контроль рабочего времени. — [Электронный ресурс]. — Режим доступа: <https://ric501.ru/wp-content/uploads/sites/29/2021/06/Statya-Elektronnyj-propusknoj-rezhim-kontrol-rabochego-vrem.pdf> (дата обращения: 23.03.24).
2. М. М. Крикунов, А. Н. Поручиков. ОСНОВЫ БАЗ ДАННЫХ. // . — 2021. — С. 84.
3. М. Клеппман. Высоконагруженные приложения. Программирование, масштабирование, поддержка. // . — 2018. — С. 640.
4. Popularity Ranking of Database Management Systems. — [Электронный ресурс]. — Режим доступа: <https://arxiv.org/pdf/2301.00847.pdf> (дата обращения: 24.03.24).
5. Проектирование информационной обучающей системы выбор СУБД и способа доступа к данным. — [Электронный ресурс]. — Режим доступа: https://www.elibrary.ru/download/elibrary_36316536_76885812.pdf (дата обращения: 24.03.24).
6. About MySQL. — [Электронный ресурс]. — Режим доступа: <https://www.oracle.com/mysql/what-is-mysql/> (дата обращения: 24.03.24).
7. Сравнение различных СУБД. — [Электронный ресурс]. — Режим доступа: https://www.elibrary.ru/download/elibrary_26575086_73187539.pdf (дата обращения: 24.03.24).
8. СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЛЯЦИОННЫХ СУБД, ПРИМЕНЯЕМЫХ ДЛЯ WEB - ПРИЛОЖЕНИЯ. — [Электронный ресурс]. — Режим доступа: https://www.elibrary.ru/download/elibrary_48375110_15267617.pdf (дата обращения: 24.03.24).
9. СРАВНЕНИЕ MYSQL и POSTGRES SQL. — [Электронный ресурс]. — Режим доступа: https://www.elibrary.ru/download/elibrary_58803813_49724862.pdf (дата обращения: 24.03.24).
10. About Microsoft SQL Server. — [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16> (дата обращения: 25.03.24).

11. Выбор системы управления базами данных для программного комплекса приемной комиссии ДонНТУ. — [Электронный ресурс]. — Режим доступа: https://www.elibrary.ru/download/elibrary_42755755_59333779.pdf (дата обращения: 25.03.24).
12. About PostgreSQL. — [Электронный ресурс]. — Режим доступа: <https://www.postgresql.org/about/> (дата обращения: 24.03.24).
13. PostgreSQL vs MySQL: The Critical Differences. — [Электронный ресурс]. — Режим доступа: <https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/> (дата обращения: 27.03.24).
14. Системы управления данными категории NoSQL. — [Электронный ресурс]. — Режим доступа: https://www.ispras.ru/publications/nosql_data_management_systems.pdf (дата обращения: 24.03.24).
15. Сравнение MongoDB и Redis. — [Электронный ресурс]. — Режим доступа: <https://www.mongodb.com/compare/mongodb-vs-redis> (дата обращения: 25.03.24).
16. СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ РЕЛЯЦИОННЫХ И НЕРЕЛЯЦИОННЫХ БАЗ ДАННЫХ НА ПРИМЕРЕ MYSQL И MONGODB. — [Электронный ресурс]. — Режим доступа: https://www.elibrary.ru/download/elibrary_29650154_63824441.pdf (дата обращения: 25.03.24).
17. A Distributed Storage and Access Approach for Massive Remote Sensing Data in MongoDB. — [Электронный ресурс]. — Режим доступа: <https://www.mdpi.com/2220-9964/8/12/533> (дата обращения: 27.03.24).
18. A COMPARISON BETWEEN NOSQL AND RDBMS: STORAGE AND RETRIEVAL. — [Электронный ресурс]. — Режим доступа: <https://www.minarjournal.com/dergi/a-comparison-between-nosql-and-rdbms-storage-and-retrieva20230620121910.pdf> (дата обращения: 27.03.24).