

УДК 004.65

UDC 004.65

5.2.2. – Математические, статистические и инструментальные методы в экономике

5.2.2 – Mathematical, statistical and instrumental methods in economics

### **РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ВИЗУАЛИЗАЦИИ И ТРЕКИНГА ПРОФЕССИОНАЛЬНЫХ НАВЫКОВ В ИТ-СФЕРЕ**

### **DEVELOPMENT OF A WEB APPLICATION FOR VISUALIZATION AND TRACKING OF PROFESSIONAL SKILLS IN THE IT SPHERE**

Авакимян Наталья Николаевна  
К.ф.–м.н., доцент  
РИНЦ SPIN–код: 6082–4770  
email: avnatali@mail.ru  
*ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13*

Avakimyan Natalia Nikolaevna  
Cand.Phys.–Math.Sci., associate professor,  
RSCI SPIN–code: 6082–4770  
email: avnatali@mail.ru  
*Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13*

Куцевалов Дмитрий Сергеевич  
студент группы ИТ2301  
email: kutdmitry@gmail.com  
*ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13*

Kutsevalov Dmitry Sergeevich  
*student of group IT 2301*  
email: kutdmitry@gmail.com  
*Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13*

В условиях роста спроса на онлайн-образование и самообучение критически важны инструменты, обеспечивающие структурированный контроль над процессом освоения навыков. Разрабатываемая база данных для платформы «Track-Stack» призвана решить эту задачу, предлагая централизованное хранение данных о пользователях, учебных роадмапах, навыках и прогрессе. Цель проекта — создать надежное full-stack решение, которое не только гарантирует целостность и безопасность данных, но и обеспечивает эффективное взаимодействие клиентской и серверной частей системы, соответствующее современным стандартам разработки. Использование современных технологий (Next.js, NestJS, Prisma) обеспечивает строгую типизацию, минимализм кода и высокую производительность. Гибридная архитектура PostgreSQL с поддержкой JSONB позволяет сочетать структурированное хранение данных с адаптивностью к изменениям бизнес-логики. Внедрение ролевой модели безопасности и механизмов шифрования (JWT, bcrypt) защищает конфиденциальную информацию, а интеграция с Big Data-инструментами открывает возможности для прогнозной аналитики и персонализации обучения. Это делает платформу «Track-Stack» готовой к масштабированию и трансформации в соответствии с трендами EdTech. Ключевые объекты системы включают пользователей, роадмапы, навыки и задачи. Пользователи делятся на три категории: студенты, преподаватели и администраторы

With the growing demand for online education and self-study, tools that provide structured control over the process of mastering skills are critically important. The database being developed for the Track-Stack platform is designed to solve this problem by offering centralized storage of data on users, educational roadmaps, skills, and progress. The goal of the project is to create a reliable full-stack solution that not only guarantees the integrity and security of data, but also ensures effective interaction between the client and server parts of the system, corresponding to modern development standards. The use of modern technologies (Next.js, NestJS, Prisma) ensures strong typing, minimal code, and high performance. The hybrid PostgreSQL architecture with JSONB support allows you to combine structured data storage with adaptability to changes in business logic. The implementation of a role-based security model and encryption mechanisms (JWT, bcrypt) protects confidential information, and integration with Big Data tools opens up opportunities for predictive analytics and personalization of training. This makes the Track-Stack platform ready for scaling and transformation in line with EdTech trends. The key objects of the system include users, roadmaps, skills and tasks. Users are divided into three categories: students, teachers and administrators

Ключевые слова: ВЕБ-ПРИЛОЖЕНИЕ,  
РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ, ТРЕКИНГ  
ПРОФЕССИОНАЛЬНЫХ НАВЫКОВ

Keywords: WEB APPLICATION, RELATIONAL  
DATABASES, PROFESSIONAL SKILLS  
TRACKING

<http://dx.doi.org/10.21515/1990-4665-212-001>

Онлайн-платформа «Track-Stack» разработана для решения актуальной задачи — структурированного управления образовательными картами в условиях роста спроса на самообучение.

Рoadмапы – это структурированные программы, такие как «Frontend-разработчик» или «Rust-разработчик». Каждый roadmap характеризуется уровнем сложности, количеством этапов и списком технологий. Навыки, входящие в состав программ, имеют кросс-функциональный характер. Например, навык «Java Script» может быть частью нескольких roadmaps, что повышает гибкость системы.

Прогресс пользователя рассчитывается на основе выполнения задач, которые делятся на теоретические и практические. Все задачи равнозначны: отметка о выполнении любой из них влияет на общий процент освоения навыка. В разделе «Мои навыки» отображаются освоенные и изучаемые компетенции, а также фокусные навыки, выделенные для углубленного изучения.

Особенностью платформы является адаптивность интерфейса. Блоки «Дополнительно», «Инструменты» и «Практика» не отображаются в списке навыков, но учитываются при расчете общего прогресса. Это обеспечивает баланс между простотой восприятия и полнотой данных.

Для реализации базы данных выбрана PostgreSQL — реляционная СУБД с открытым исходным кодом, поддерживающая принципы ACID. Её преимущества включают надежность транзакций, масштабируемость и возможность работы с JSONB-полями, что позволяет совмещать структурированное хранение данных с гибкостью NoSQL. Например, метаданные о поведении пользователей или динамические параметры

<http://ej.kubagro.ru/2025/08/pdf/01.pdf>

курсов хранятся в JSONB, обеспечивая адаптивность к изменениям бизнес-логики.

Взаимодействие с базой данных организовано через Prisma ORM. Этот инструмент автоматизирует генерацию типобезопасных моделей, исключая рассогласование между кодом и схемой БД. Миграции управляются декларативно, что упрощает изменение структуры данных в рабочих средах. Например, добавление индексов для ускорения поиска или модификация типов полей выполняется без простоев.

Фронтенд-часть реализована на Next.js – фреймворке, обеспечивающем серверный рендеринг (SSR) и SEO-оптимизацию. Для управления состоянием приложения используются Zustand и TanStack Query, которые оптимизируют кэширование данных и улучшают отзывчивость интерфейса.

Ролевая модель (RBAC) разделяет права доступа: студенты могут редактировать только свой прогресс, преподаватели — управлять учебными материалами, администраторы — контролировать всю систему.

Инфологическая модель приведена на рисунке 1.

Сущности и атрибуты приведены в таблице 1.

Таблица 1- Таблица сущностей и атрибутов

Сущность	Атрибуты
User	userId (PK), email, password
Skill	skillId (PK), name
Roadmap	roadmapId (PK), title, complexity, stages, technologies
Task	taskId (PK), title, skillId (FK)
UserSkill	userId (FK), skillId (FK), progress
UserRoadmap	userId (FK), roadmapId (FK), progress
UserTask	userId (FK), taskId (FK), completed

Инфологическая модель отражает взаимосвязи между сущностями системы. Таблица User хранит данные пользователей, включая email и

хешированный пароль. Каждый пользователь связан с роадмапами через UserRoadmap, где фиксируется общий прогресс выполнения программы.

Рoadмапы (Roadmap) состоят из навыков (Skill), объединенных через связующую таблицу. Навыки содержат задачи (Task), привязанные через внешний ключ skillId. Прогресс по навыкам регистрируется в UserSkill, а выполнение задач — в UserTask. Отдельная таблица FocusSkill дублирует структуру UserSkill, но служит для выделения приоритетных компетенций в интерфейсе.

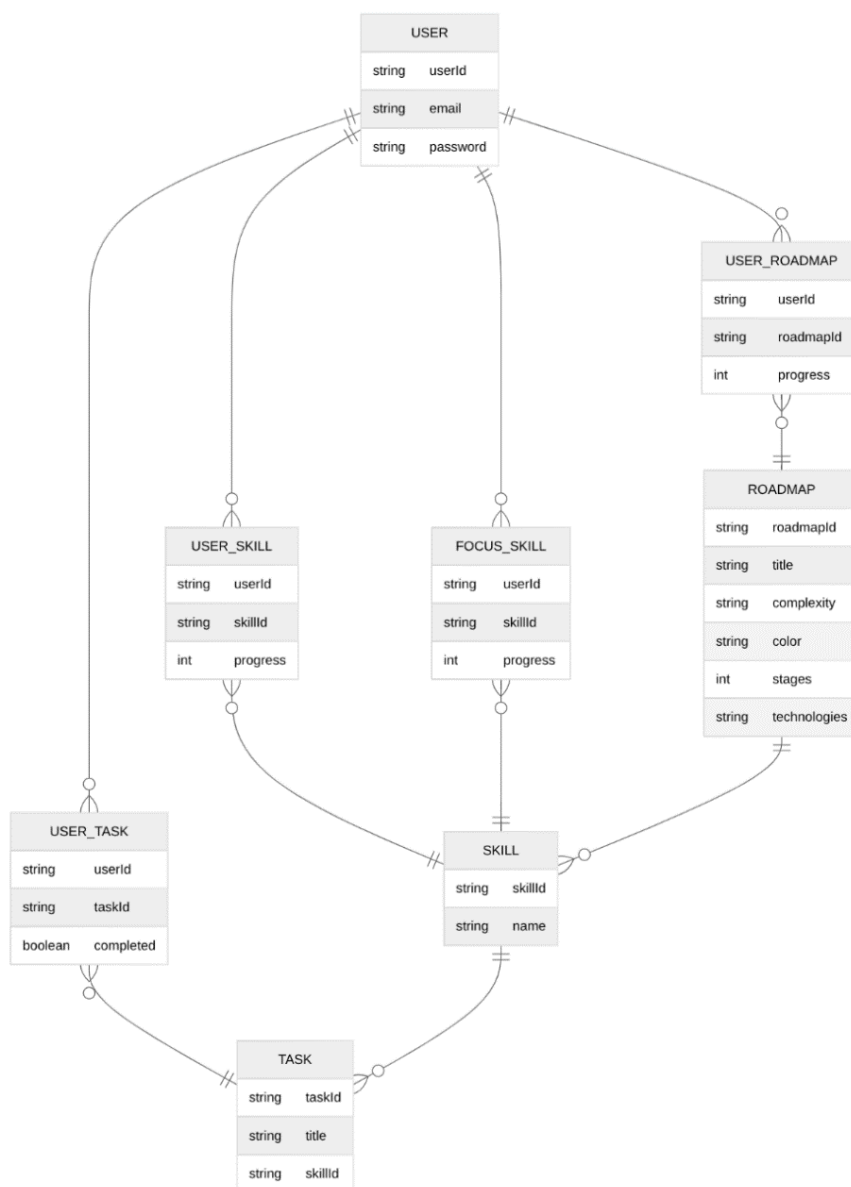


Рисунок 1 – Инфологическая модель

Схема обеспечивает целостность данных через внешние ключи и ограничения. Например, поле `progress` в `UserSkill` проверяется на допустимый диапазон (0–100%), а каскадное удаление гарантирует очистку зависимых записей.

ER-диаграмма модели включает сущности в виде прямоугольников с атрибутами и связи с указанием мощности. Например, связь между `User` и `Skill` через `UserSkill` имеет атрибут `progress`, а связь `User` и `Task` через `UserTask` — атрибут `completed`.

Модель поддерживает основные бизнес-процессы: формирование роадмапов, учет прогресса и управление учебными материалами. Её гибкость позволяет масштабировать систему, добавляя новые сущности (например, тесты или сертификаты) без нарушения существующей логики.

Проектирование базы данных для платформы «Track-Stack» осуществлялось с использованием PostgreSQL 15, что обеспечило соблюдение принципов ACID и поддержку гибридного хранения данных. Основой структуры стала инфологическая модель, описанная во второй главе, включающая сущности пользователей, навыков, роадмапов и задач. Для автоматизации работы с БД был задействован Prisma ORM, позволяющий генерировать типобезопасные модели на основе декларативной схемы. Миграции выполнялись через инструменты Prisma, что обеспечило бесшовное внесение изменений в рабочую среду, таких как добавление индексов для оптимизации поиска или модификация типов полей.

```
generator client {  
  provider = "prisma-client-js"  
}  
datasource db {  
  provider = "postgresql"  
  url      = env("DATABASE_URL")  
}  
model User {
```

```
    userId    String    @id @default(cuid())
    email     String    @unique
    password  String
    userRoadmaps UserRoadmap[]
    userSkills UserSkill[]
    focusSkills FocusSkill[]
    userTasks  UserTask[]
}
model Skill {
    skillId    String    @id @default(cuid())
    name       String    @unique
    userSkills  UserSkill[]
    focusSkills FocusSkill[]
    tasks      Task[]
    roadmaps   Roadmap[] @relation("RoadmapToSkill")
}
model Roadmap {
    roadmapId String    @id
    title     String
    complexity String
    color     String
    stages    Int
    technologies String
    userRoadmaps UserRoadmap[]
    skills     Skill[] @relation("RoadmapToSkill")
}
model UserSkill {
    userId String
    skillId String
    progress Int
    user   User @relation(fields: [userId], references: [userId])
    skill  Skill @relation(fields: [skillId], references: [skillId])

    @@id([userId, skillId])
}
model FocusSkill {
    userId String
    skillId String
```

```
progress Int
user User @relation(fields: [userId], references: [userId])
skill Skill @relation(fields: [skillId], references: [skillId])

@@id([userId, skillId])
}

model UserRoadmap {
  userId String
  roadmapId String
  progress Int
  user User @relation(fields: [userId], references: [userId])
  roadmap Roadmap @relation(fields: [roadmapId], references: [roadmapId])
  @@id([userId, roadmapId])
}

model Task {
  taskId String @id @default(cuid())
  title String
  skillId String
  skill Skill @relation(fields: [skillId], references: [skillId])
  userTasks UserTask[]
}

model UserTask {
  userId String @map("user_id")
  taskId String @map("task_id")
  completed Boolean @default(false)
  user User @relation(fields: [userId], references: [userId])
  task Task @relation(fields: [taskId], references: [taskId])

  @@id([userId, taskId])
  @@map("user_tasks")
}
```

Взаимодействие таблиц в базе данных платформы «Track-Stack» организовано через систему связей, отражающих логику образовательного процесса. Основная сущность – User – связывается с роадмапами (Roadmap) и навыками (Skill) через промежуточные таблицы UserRoadmap и UserSkill, что позволяет фиксировать прогресс пользователя по каждому

направлению. Например, таблица UserRoadmap хранит данные о том, насколько продвинулся студент в изучении конкретной программы, а UserSkill отслеживает освоение отдельных компетенций. Навыки, выделенные для углубленного изучения, регистрируются в FocusSkill, которая наследует структуру UserSkill, но служит для визуального выделения в интерфейсе.

Рoadмапы (Roadmap) объединяют группы навыков, формируя структурированные учебные программы. Каждый навык (Skill) связан с задачами (Task), которые пользователи отмечают, как выполненные через таблицу UserTask. Это обеспечивает детализацию прогресса: выполнение задачи автоматически влияет на общий процент освоения навыка, а через него — на прогресс в roadмапе. Связи между сущностями реализованы через внешние ключи, гарантирующие целостность данных. Например, удаление навыка из системы приводит к каскадному удалению связанных с ним задач и записей в UserSkill, что исключает возникновение «висячих» ссылок.

Архитектура БД поддерживает гибкость: JSON-поля в Roadmap позволяют хранить динамические параметры, такие как цветовое кодирование или список технологий, а использование составных первичных ключей в связующих таблицах предотвращает дублирование записей. Такая структура обеспечивает масштабируемость системы, позволяя добавлять новые сущности (например, тесты или сертификаты) без нарушения существующей бизнес-логики.

Архитектура приложения реализована по принципу разделения на клиентскую и серверную части. Фронтенд-часть, построенная на Next.js, обеспечила серверный рендеринг и адаптивный интерфейс, интегрирующий визуализацию прогресса и управление учебными планами. Главная страница, представленная в разделе «Построй свой путь в IT», начинается с мотивационного призыва «Стань востребованным



разработчиком», после чего пользователю предлагается выбор популярных направлений: Frontend-разработка, Backend-разработка, Искусственный интеллект, DevOps и Data Science (рисунок 2). Каждое направление оформлено в виде интерактивного элемента, позволяющего начать отслеживание прогресса.

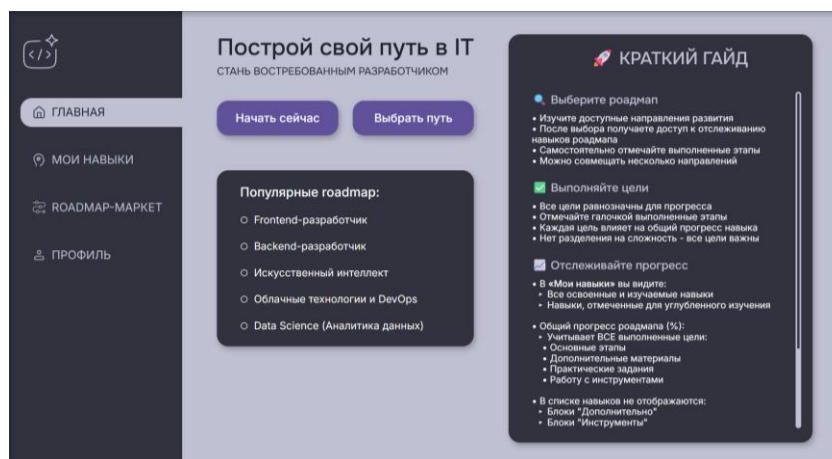


Рисунок 2 – Главная страница

Ключевой раздел «Краткий гайд» структурирует процесс работы с платформой. На первом этапе пользователь изучает доступные роадмапы и выбирает интересующее направление, после чего получает доступ к детализированному списку навыков и задач. Все цели, включая теоретические материалы, практические задания и работу с инструментами, равнозначны для общего прогресса — выполнение любой из них автоматически обновляет процент освоения навыка. Это обеспечивает гибкость: пользователь может фокусироваться на задачах, соответствующих его текущему уровню, без строгой привязки к сложности.

Страница «Мои навыки» предоставляет пользователю детализированный обзор его образовательных достижений. В центральном блоке отображаются освоенные и текущие навыки, такие как HTML, CSS и JavaScript, с указанием прогресса в процентах. Например, HTML отмечен

как полностью освоенный (100%), а CSS и JavaScript находятся на стадии активного изучения (67%). Фокус-навыки выделены в отдельный раздел, позволяя пользователю акцентировать внимание на приоритетных компетенциях. Это реализовано через связь с таблицей FocusSkill в базе данных, где хранятся выбранные для углубленного изучения навыки. Управление фокус-навыками осуществляется через кнопки «Добавить» и «Удалить», которые запускают соответствующие запросы к API, обновляя данные в реальном времени.

В блоке «Активные roadmap» отображается прогресс по выбранным программам. Например, roadmap «Frontend-разработчик» завершен на 75%, что визуализируется через индикатор выполнения. Прогресс уровня пользователя (80%) рассчитывается на основе совокупности завершенных задач и навыков, что мотивирует к дальнейшему обучению. Подсказка «Вы на правильном пути!» добавляет элемент геймификации, повышая вовлеченность.

Интерфейс интегрирован с клиентской логикой приложения: данные о навыках и прогрессе загружаются асинхронно через TanStack Query, что минимизирует задержки. Состояние элементов, таких как активные roadmap или фокус-навыки, управляется через Zustand, обеспечивая синхронизацию между компонентами без перезагрузки страницы. Поиск навыков реализован с использованием дебаунсинга, чтобы снизить нагрузку на сервер при частых запросах. Адаптивный дизайн страницы, созданный с помощью Tailwind CSS, гарантирует корректное отображение на мобильных устройствах и планшетах. Взаимодействие с элементами интерфейса, например отметка выполнения задачи, мгновенно отражается в базе данных через вызовы Prisma Client на сервере, сохраняя целостность данных. Это создает seamless-опыт, где пользователь видит результаты своих действий без задержек, а система остается отзывчивой даже при высокой нагрузке.

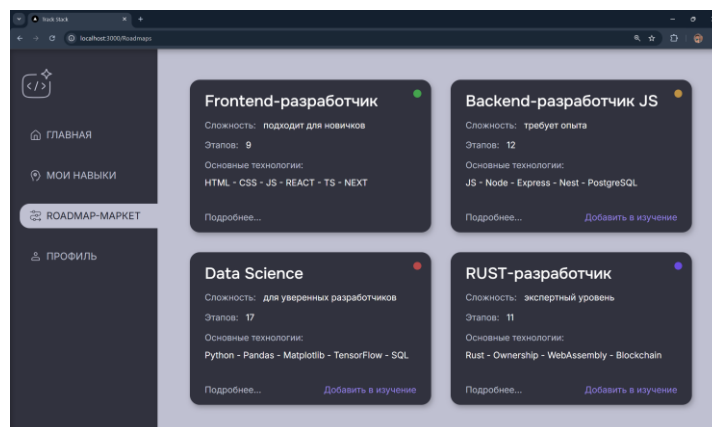


Рисунок 3 – Roadmap маркет

Страница выбора роадмапов, такая как «Frontend-разработчик» или «Data Science», предоставляет пользователю структурированный обзор доступных образовательных программ. Каждый роадмап содержит метаданные: уровень сложности (например, «новичок» или «эксперт»), количество этапов, список ключевых технологий и кнопку «Добавить в изучение». Данные о роадмапах хранятся в таблице Roadmap базы данных, где поля complexity, stages и technologies заполняются на этапе администрирования. При клике на «Подробнее...» пользователь переходит на детализированную страницу роадмапа, которая динамически загружается через API-запрос к серверу, использующему Prisma для извлечения данных из PostgreSQL.

Детализированный роадмап, например «Frontend Roadmap», разбит на разделы: HTML, CSS, JavaScript, инструменты и практические задания. Каждый раздел включает список задач с отметками о выполнении (чекбоксы), которые синхронизируются с таблицей UserTask через вызовы к NestJS-серверу. Прогресс по этапам автоматически пересчитывается и отображается в виде процентов, используя агрегацию данных через Prisma Client. Например, завершение всех задач в разделе «HTML» увеличивает общий прогресс роадмапа, что отражается как в интерфейсе, так и в связанных таблицах UserRoadmap и UserSkill.

Кнопка «Добавить в изучение» инициирует создание записи в UserRoadmap, связывая пользователя с выбранной программой. Это действие выполняется через REST API с использованием JWT-токена для аутентификации, гарантируя, что только авторизованные пользователи могут изменять свои учебные планы. Состояние интерфейса, включая активные роадмапы и выполненные задачи, управляется через Zustand, что обеспечивает мгновенное обновление данных без перезагрузки страницы.

Адаптивность интерфейса достигается за счёт использования компонентов Next.js и Tailwind CSS. Например, блоки с технологиями (HTML, CSS, React) автоматически подстраиваются под размер экрана, а интерактивные элементы (чекбоксы, кнопки) оптимизированы для сенсорных устройств. Интеграция с TanStack Query минимизирует задержки при загрузке данных: кэширование списка роадмапов и деталей программ снижает нагрузку на сервер и улучшает пользовательский опыт.

Таким образом, страницы выбора и детализации роадмапов демонстрируют тесную интеграцию клиентской и серверной частей приложения. Использование современных инструментов (Prisma, Zustand, Next.js) обеспечивает не только высокую производительность, но и гибкость, позволяя платформе адаптироваться к новым образовательным трендам и требованиям пользователей.

Разработанная платформа «Track-Stack» представляет собой полноценное full-stack решение для трекинга профессиональных навыков в IT-сфере, объединяющее современные технологии и методологии управления данными.

#### **Список использованной литературы**

1. Гарсиа-Молина, Г. Базы данных. Полный курс [Электронный ресурс]: учебник / Г. Гарсиа-Молина, Д. Ульман, Дж. Уйдом. — Базы данных. Полный курс, 2023-05-10. — Москва: Лань, 2023. — 896 с.
2. Таненбаум, Э. Распределенные системы. Базы данных и обработка данных: учебник / Э. Таненбаум, М. ван Стеен. — 3-е изд. — СПб.: Питер, 2022. — 560 с.

3. Дейт, К. Дж. Введение в системы баз данных [Электронный ресурс]: учебное пособие / К. Дж. Дейт. — Введение в системы баз данных, 2023-03-15. — М.: ДМК Пресс, 2023. — 780 с.

### References

1. Garcia-Molina, G. Databases. Complete Course [Electronic resource]: textbook / G. Garcia-Molina, D. Ullman, J. Widom. — Databases. Complete Course, 2023-05-10. — Moscow: Lan, 2023. — 896 p.

2. Tanenbaum, E. Distributed Systems. Databases and Data Processing: textbook / E. Tanenbaum, M. van Steen. — 3rd ed. — St. Petersburg: Piter, 2022. — 560 p.

3. Date, K. J. Introduction to Database Systems [Electronic resource]: textbook / K. J. Date. — Introduction to Database Systems, 2023-03-15. — Moscow: DMK Press, 2023. — 780 p.