ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

DOI: 10.32517/0234-0453-2024-39-5-21-39

СОВРЕМЕННЫЕ ПОДХОДЫ К ОРГАНИЗАЦИИ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА СБОРА И ПОДГОТОВКИ БОЛЬШИХ ДАННЫХ И СВЯЗАННЫЕ С НИМИ НОВЫЕ ПРОФЕССИИ В ИТ-ОТРАСЛИ: DEVOPS-ИНЖЕНЕР И ИНЖЕНЕР ДАННЫХ

Н. Н. Самылкина[1] ✉, **И. А. Калинин**[2]

[1] ***Московский педагогический государственный университет, г. Москва, Россия***

[2] ***Московский государственный лингвистический университет, г. Москва, Россия***

✉ *nsamylkina@yandex.ru*

Аннотация

В статье рассмотрены современные технологические подходы к организации сбора и обработки больших данных, а также распространения и использования приложений с помощью платформы Docker с загрузкой и запуском комплекса приложений на основе контейнерного подхода к виртуализации.

Кратко изложены основы постановки задачи по обработке больших данных и возникающие при этом технические проблемы масштабирования технической платформы и обеспечения качества данных. На примере развертывания и использования системы управления базами данных PostgreSQL со средствами управления и аналитической платформы Apache Superset представлен современный подход к организации получения, трансформации и последующей демонстрации данных в виде пульта. Работа с данными показана на примере цикла обработки реального набора открытых данных с демонстрацией всех основных этапов — загрузки, очистки, предварительного анализа и визуализации.

В контексте рассмотренных технологий и подходов к организации рабочего процесса в виде конвейера приводятся примеры рабочих задач DevOps-инженера и инженера данных, подчеркивается необходимость их работы в составе команды.

Продемонстрированные в статье технические решения могут также применяться при изучении тем «Базы данных», «Инфографика», «Анализ данных» в углубленном курсе информатики технологического профиля в качестве иллюстрации сложности рассматриваемых процессов и при подготовке будущих учителей информатики.

Ключевые слова: большие данные, ETL-конвейер, виртуализация, масштабирование Docker, Apache Superset, визуализация, анализ данных, DevOps-инженер, инженер данных.

Для цитирования:

Самылкина Н. Н., Калинин И. А. Современные подходы к организации технологического процесса сбора и подготовки больших данных и связанные с ними новые профессии в ИТ-отрасли: DevOps-инженер и инженер данных. *Информатика и образование.* 2024;39(5):21–39. DOI: 10.32517/0234-0453-2024-39-5-21-39.

MODERN APPROACHES TO THE ORGANIZATION OF THE TECHNOLOGICAL PROCESS OF BIG DATA COLLECTION AND PREPARATION AND RELATED NEW PROFESSIONS IN THE IT INDUSTRY: DEVOPS-ENGINEER AND DATA ENGINEER

N. N. Samylkina[1] ✉, **I. A. Kalinin**[2]

[1] ***Moscow Pedagogical State University, Moscow, Russia***

[2] ***Moscow State Linguistic University, Moscow, Russia***

✉ *nsamylkina@yandex.ru*

Abstract

The article modern technological approaches to the organization of big data collection and preparation, as well as the distribution and use of applications with the help of the Docker platform, loading and launching a set of applications on the basis of the container approach to virtualization, are considered.

The basics of big data processing task formulation and technical problems of technical platform scaling and data quality assurance are briefly outlined. On the example of deploying and using PostgreSQL database management system with management tools and Apache Superset analytics platform, a modern approach to organizing data acquisition, transformation and subsequent demonstration in the form of a dashboard is presented. Work with data is shown on the example of a real open data set processing cycle with demonstration of all the main stages — loading, cleaning, preliminary analysis and visualization.

In the context of the considered technologies and approaches to workflow organization in the form of a pipeline, examples of DevOps engineer and data engineer work tasks are given, the necessity of their work as part of a team is emphasized.

The technical solutions demonstrated in the article can also be used in the study of the topics “Databases”, “Infographics”, and “Data analysis” in the advanced informatics course of technological profile as an illustration of the complexity of the relevant processes and in the training of future informatics teachers.

Keywords: big data, ETL pipeline, virtualization, Docker scaling, Apache Superset, visualization, data analysis, DevOps engineer, data engineer.

For citation:

Samylkina N. N., Kalinin I. A. Modern approaches to the organization of the technological process of big data collection and preparation and related new professions in the IT industry: DevOps-engineer and data engineer. *Informatics and Education.* 2024;39(5): 21–39.(In Russian.) DOI: 10.32517/0234-0453-2024-39-5-21-39.

1. Актуальность темы

Данных вокруг нас становится все больше и больше. Отчет аналитической компании DOMO демонстрирует, что количество данных, которые генерировали пользователи в наиболее популярных сервисах каждую минуту, возросло с 2013 года к 2023 году в несколько раз, а в некоторых сервисах в десять раз и более[1]. Заметим, что в этом отчете не были проанализированы внутренние данные компаний, данные «интернета вещей» и др., а их количество также нарастает с каждым днем и даже минутой.

Аналогичные выводы делают и другие информационно-аналитические агентства и сервисы. Среди поисковых запросов Google понятия, связанные с данными: «наука о данных» (*англ.* Data Science), «аналитика данных» (*англ.* Data Analytics), «визуализация данных» (*англ.* Data Visualization) и т. п., — встречались последние двадцать лет все чаще. Согласно данным сервиса Google Trends, в феврале и мае 2024 года, например, популярность запроса «Data Science» достигала своего пика (рис. 1).

Для обработки данных требуются не только специализированные технологии и инструменты, но и профессионалы, которые ими эффективно владеют. Количество вакансий в наиболее востребованных направлениях профессиональной деятельности в сфере больших данных (аналитик данных (*англ.* Data Analyst), инженер данных (*англ.* Data Engineer), дата-сайентист (*англ.* Data Scientist) и др.) также постоянно растет (рис. 2).

[1] *McLean J.* Data never sleeps turn 11! *Domo Blog.* 23.12.2023. https://www.domo.com/blog/data-never-sleeps-turns-11/

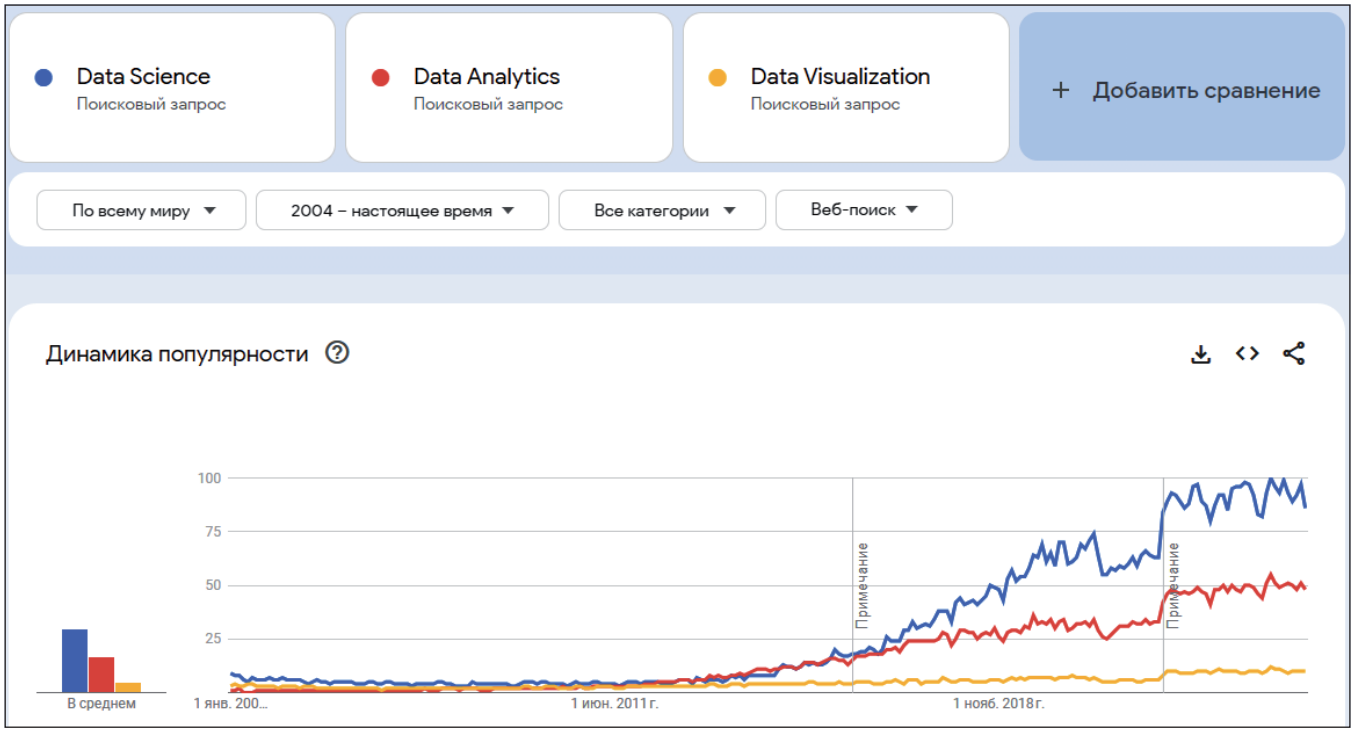

Рис. 1. Динамика популярности поисковых запросов Google с 2004 года по 2024 год

Fig. 1. Dynamics of mentions in Google search queries from 2004 to 2024

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

Эта тенденция сохранялась и в 2023 году: на фоне общей нехватки профессионалов в области информационных технологий популярность этого направления на рынке труда не снижается (рис. 3).

Обновленный в 2023 году Федеральный государственный образовательный стандарт среднего общего образования в требованиях к освоению предмета «Информатика» включает в себя лишь отдельные

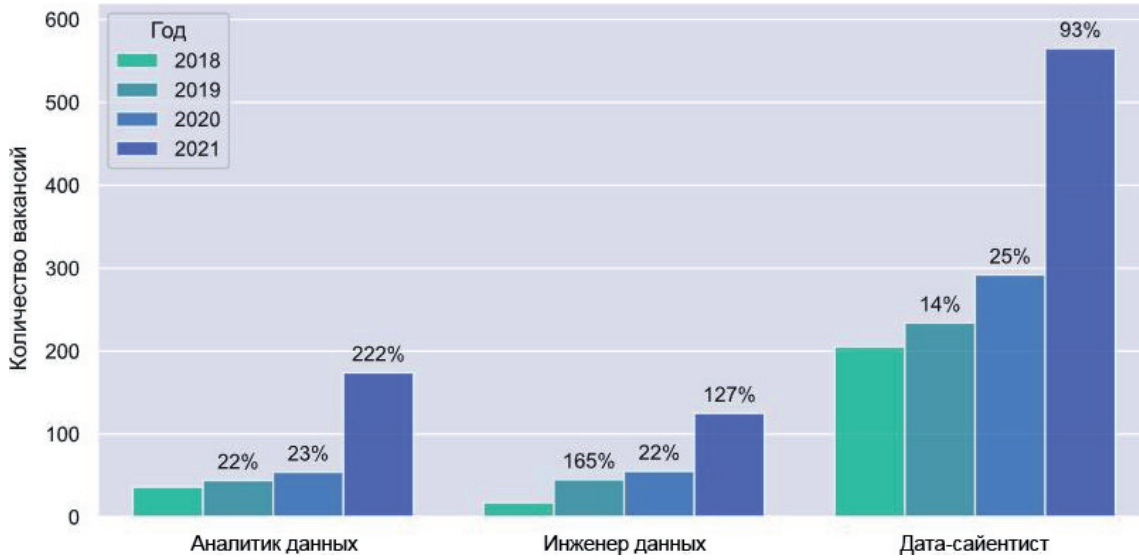

Рис. 2. Динамика количества вакансий в сфере больших данных за 2018–2021 годы[1]

Fig. 2. Dynamics of the number of vacancies in the field of big data from 2018 to 2021

Сегмент IT	Позиция	Россия	Москва	Санкт-Петербург	Новосибирск	Екатеринбург	Казань	Другие регионы
Разработка ПО	Тестировщик	110 571	29 765	17 967	3 404	2 581	3 029	53 825
	Frontend-разработчик	52 513	16 845	8 542	1 245	1 400	1 427	23 054
	Python-разработчик	35 119	11 861	5 727	974	1 063	986	14 508
	Backend-разработчик	16 427	5 136	2 733	431	406	500	7 221
	Fullstack-разработчик	5 018	1 592	801	139	111	119	2 256
Аналитика и данные	Аналитик данных	26 081	12 893	4 378	520	548	537	7 205
	Data Scientist	13 058	7 155	2 079	352	306	260	2 906
	Маркетолог-аналитик	5 066	2 088	690	114	165	102	1 907
	Веб-аналитик	1 079	550	157	25	16	16	315
Продакт и проджект	Проджект-менеджер	279 605	138 485	37 463	4 853	6 698	5 377	86 729
	Продакт-менеджер	41 505	27 539	5 789	699	725	469	6 284
Дизайн	Графический дизайнер	68 132	19 036	9 383	1 645	2 039	1 494	34 535
	Web-дизайнер	29 188	7 817	3 995	716	771	753	15 136
	UX/UI-дизайнер	23 326	9 044	4 473	541	603	599	8 066
	Геймдизайнер	7 010	2 346	1 316	354	145	89	2 760
	Motion-дизайнер	4 170	1 523	911	91	103	107	1 435
IT инфраструктура	Системный администратор	135 545	30 403	12 748	3 208	2 768	2 703	83 715
	Системный инженер	14 878	3 902	1 844	298	356	268	8 210
	Сетевой инженер	9 354	3 278	1 090	262	231	193	4 300
	Сервисный инженер	2 717	546	243	72	58	48	1 750

hh 2023 год

Рис. 3. Количество резюме в области информационных технологий в 2022–2023 годах (по данным компании HeadHunter[2]*)*

Fig. 3. Number of IT CVs in 2022–2023 (according to HeadHunter)

[1] *Борисов Е.* Анализ вакансий и зарплат в Data Science. *Хабр.* 26.08.2021. https://habr.com/ru/companies/ods/articles/572264/

[2] *Кравченко К.* Обзор рынка труда в ИТ: III квартал 2023 года. *Proglib.* 08.12.2023. https://proglib.io/p/obzor-rynka-truda-v-it-iii-kvartal-2023-goda-2023-12-08

ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2024 • Том 39 № 5

знания и умения в рамках перечисленных выше технологических направлений, причем только в случае углубленного изучения этого предмета в X—XI классах, в частности [1]:

- *при изучении раздела «Теоретические основы информатики»:*
  - «умение классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений)»;
  - «умение понимать последовательность решения задач анализа данных (сбор первичных данных, очистка данных и оценка их качества, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов)»;
- *при изучении раздела «Информационные технологии»:*
  - «владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними»;
  - «умение использовать табличные (реляционные) базы данных (составлять запросы в базах данных, выполнять сортировку и поиск записей в базе данных, наполнять разработанную базу данных) и справочные системы».

Приходится констатировать, что пока за рамками школьного стандарта остаются современные технологические подходы к накоплению данных и к организации их обработки, а также необходимость подготовки и сопровождения этого технологического процесса. Причина заключается в сложности самих задач и используемого программного обеспечения. В целом из тематического раздела «Информационные технологии» в курсе «Информатики» выпадают информационные системы в качестве самостоятельного предметного результата, в значительной степени влияющего на выбор профессиональной траектории в информационно-технологической отрасли.

Вполне очевидно, что в дальнейшем информатика как учебная дисциплина будет развиваться в направлении интеллектуальных информационных систем. Для этого имеются определенные предпосылки, например, обновление образовательных стандартов во многих зарубежных странах [2, 3] на фоне повышенного интереса к вопросам организации сбора, обработки и анализа данных для различных областей профессиональной деятельности [4, 5].

Вместе с тем в нашей стране накоплен большой практический опыт изучения основ искусственного интеллекта и анализа данных в курсе информатики, что отражено в научных публикациях (см., например, [6, 7]). Появляются новые методические разработки, посвященные изучению популярных технологий искусственного интеллекта в раннем возрасте [8].

В ходе научно-методической работы по существенному обновлению материала школьного курса информатики, предлагаемых в этом курсе учебных задач и применяемых технических средств необходимо помнить о профессиональном самоопределении старшеклассников. Требуется информировать учащихся о современных профессиях, демонстрировать профессиональные задачи и используемые программные продукты. Начинать следует с будущих учителей информатики, которые придут в современную школу и будут преподавать обновленный курс информатики.

Целью статьи является раскрытие на демонстрационном примере современных подходов к организации технологического процесса сбора и подготовки больших данных для их дальнейшего использования в любой производственной отрасли. Таким образом можно показать различие в профессиональных навыках DevOps-инженера и инженера данных, которые необходимо учитывать выпускникам при выборе направления профессиональной подготовки.

2. «Большие данные»: истоки появления понятия

Какие навыки нужны, чтобы посчитать среднее значение изменяющейся величины, например, для ответа на вопрос: «сколько раз в среднем обращаются к нашему сайту?» Если с этим вопросом обратиться к администратору сайта, то он назовет данные счетчика (например, Яндекс Метрика), среднее значение которых легко посчитать. Но откуда взялись числа в этом аналитическом сервисе? Можем ли мы посчитать их сами, например, среднее значение не за день, а за час, и получить их сразу, как только час закончился? А сколько раз обращаются к сайту в минуту? Разумеется, ответы будут в значительной мере зависеть от того, насколько сайт большой и популярный.

Рассмотрим ситуацию, в которой речь идет не просто о сайте, а об огромном сервисе (информационной системе) с десятками тысяч посетителей в секунду. Очевидно, что мы сразу столкнемся с огромным потоком данных. При этом поток данных будет продолжать расти, и даже формально небольшой относительный прирост количества пользователей будет означать огромный абсолютный рост фактического количества данных. *Данные будут поступать очень быстро.* Большое количество пользователей постоянно что-то делает на сайте, и необходимо следить как минимум за тем, чтобы сервис работал. *Данные будут очень разнообразными,* поэтому необходимо фиксировать не только факт обращения пользователя к сайту, но и время, цель обращения и то, насколько оно было удачным. При необходимости можно выделить массу параметров, по которым нужно собирать данные.

Подобные задачи и запросы создают огромное количество технических проблем. Решить их «раз и навсегда» не сможет никто, потому что в каждом конкретном случае они связаны с определенными сервисами, в которых необходимо учитывать цели обработки данных и то, что эти цели будут со временем заметно меняться. Для решения подобных задач

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

требуются особые средства, а также специалисты определенной квалификации.

Понимание того, что мы входим в совершенно новую технологическую эпоху, сформировалось в начале 2000-х годов и окончательно оформилось в статье C. Lynch «How do your data grow?», которая вышла в журнале Nature 3 сентября 2008 года [9]. В ней автор обобщил материалы о взрывном росте количества и многообразия данных в нескольких отраслях науки и промышленности и ввел (по другой информации, удачно использовал) специальный термин «большие данные» (*англ.* Big Data) [10] как общее название для задач, которые порождают поток данных, имеющий три основных признака[1]:

- объем (*англ.* Volume);
- скорость (*англ.* Velocity);
- разнообразие (*англ.* Variety).

Специфика понятия «большие данные» такова, что определенных границ этих данных никто установить не может. Какой именно объем они имеют? Какую именно скорость? Насколько данные разнообразны? Специалисты используют условные границы: например, более 100 Мбайт в секунду. Но **фактически под большими данными понимают такой объем информации, который не удастся разместить на одном (пусть даже очень мощном) сервере, который будет постоянно модернизироваться.** Это означает, что невозможно предсказать, какое количество оборудования потребуется для хранения/обработки информации, или невозможно купить его сразу, и неизвестно, какие закономерности и как следует искать для такого прогнозирования (потому что новые задачи появляются постоянно).

Большие данные генерируются в социальных сетях, при работе современных производственных линий, в медицине, в научных исследованиях, в управлении городами и других областях. Накопив опыт работы с ними, разработав методы и программные средства, а главное, найдя и обучив нужных специалистов, мы получаем массу возможностей для углубления своих представлений о сложных объектах, диагностике, принятии решений и прогнозе развития многих систем.

Работа с такими данными критически важна, но какие специалисты занимаются данными и их обработкой? Узнаем это на примере создания небольшого приложения, которое также поможет увидеть некоторые способы и приемы работы с большими данными.

3. Постановка демонстрационной проблемы

Люди очень серьезно относятся к покупке/продаже жилой недвижимости, поскольку часто это самое крупное приобретение в их жизни. С этим событием связано очень много самых разных аналитических задач и вопросов, причем не только оперативных (например, остались ли средства для обеспечения льготной ставки ипотечного кредита, выделенные государством в этом году) или аналитических (например, какие квартиры пользуются наибольшим спросом сейчас и как этот спрос изменится через три года), но и прогнозных интеллектуальных задач (в том числе задач машинного обучения). Данных для таких задач используется очень много [11, 12], они поступают быстро и вполне соответствуют термину «большие данные». Решение таких задач требует создания целого комплекса работы с данными, который строят, поддерживают и сопровождают специалисты множества современных направлений профессиональной деятельности. Мы возьмем для примера уже обработанные данные, опубликованные для открытого конкурса, и покажем, очень кратко и упрощенно, как с ними работают специалисты в области больших данных [13, 14].

Начнем, как ни странно, не с самих данных, а с основы для их накопления и дальнейшей работы с ними — с инфраструктуры самого приложения.

Как уже сказано, мы не знаем, сколько нам в итоге потребуется ресурсов, но точно знаем — все больше и больше. Поэтому инфраструктура и для работы, и для приложения должна быть рассчитана на постоянное расширение и изменение. Более того, у нас не будет возможности останавливать работу, чтобы обновить приложение, выполнить техническое обслуживание оборудования или программного обеспечения: остановка даже на короткий срок будет означать большие потери, потому что более миллиона человек потеряют по одному часу рабочего времени, а это уже более 100 человеко-лет.

Свойство, которое позволяет нам наращивать мощности на ходу, добавляя новые физические сервера для обеспечения функционирования системы, называется горизонтальным масштабированием. Например, слишком большую базу данных (БД) можно разделить на несколько серверов, но система управления базами данных (СУБД) должна поддерживать такие возможности, т. е. обеспечивать обращение к такой разделенной БД как одной точке.

Кроме того, очень часто приложения разрабатывают не с нуля, а используя как можно больше готовых компонентов. Наше приложение сетевое, и компоненты для таких приложений тоже очень часто являются сетевыми приложениями, оформленными в виде готового сервера. **Одной из технологий, которая позволяет получать готовые приложения и в дальнейшем их поддерживать, является популярная современная технология Docker** [15].

Принцип работы Docker — управление контейнерами, т. е. готовыми виртуальными серверами. В отличие от классической виртуальной машины, эта среда не эмулирует полный компьютер, а создает виртуальную копию операционной системы и позволяет добавить к ней все необходимые программы, библиотеки и настройки, т. е. создать среду для одного приложения. Такой контейнер требует гораз-

[1] *Laney D.* 3D Data Management: Controlling Data Volume, Velocity, and Variety. *Application Delivery Strategies.* Stamford, CT, USA, META Group Research Note, 2001;6:949. https://studylib.net/doc/8647594/3d-data-management--controlling-data-volume--velocity--an

до меньше ресурсов, быстрее запускается и быстрее развертывается.

Основными компонентами в среде Docker являются:

- докер-образ (*англ.* Docker Image) — шаблон, по которому создают контейнеры;
- докер-файл (*англ.* Dockerfile) — инструкция по сборке образа; как правило, из внешнего источника скачивается базовый образ, а затем создается дополнительный файл (докер-файл) для того, чтобы шаблонный образ стал образом уже нашей инфраструктуры;
- докер-контейнер (*англ.* Docker Container) — сущность на основе образа, которая выполняет собственно рабочие функции, приложение, которое мы развернули с помощью Docker;
- докер-репозиторий (*англ.* Docker Registry) — хранилище, в котором хранятся докер-образы; оно может быть как локальным, так и публичным. Репозитории создают на платформах вроде Docker Hub и GitLab и размещают в них образы с описанием, версиями и тегами.

Если требуется новое приложение, чтобы присоединить его к нашей системе, необходимо скачать образ-шаблон, написать инструкцию по его адаптации и на его основе запустить контейнер, который сразу станет доступен остальным компонентам как сетевой компьютер.

Если требуется горизонтальное масштабирование, необходимо создать еще один контейнер на основе нашего образа, запустить его (все настройки уже заданы, и приложение, конечно, должно быть написано с учетом такой работы). На практике подобные системы требуют дополнительной инфраструктуры [16], но мы рассматриваем только ее локальную часть.

Попробуем развернуть такое приложение для дальнейших иллюстраций.

4. Практическое решение демонстрационной проблемы

Важно понимать, что нам необходимо выбрать такую операционную систему (ОС), для которой есть готовые образы. Проще всего выбрать популярный дистрибутив Linux — для него нужные образы, скорее всего, найдутся. Будем считать, что у нас уже есть готовый к работе компьютер со свободно распространяемой ОС Linux Debian 12 (на момент написания статьи это стабильная версия).

Запустим терминал и начнем установку необходимого программного обеспечения.

Сначала установим программу — текстовый браузер cURL:

```
apt install curl -y
```

Теперь установим собственно Docker. Сценарий его установки довольно длинный, но он для нас уже написан, и мы можем получить его в виде текста (программой curl) и сразу построчно выполнить:

```
curl -sSL https://get.docker.com/ | sh
```

Мы могли бы управлять Docker из командной строки, но такая работа требует практики. Немного упростим себе задачу, установим сервис для управления. Он, конечно, собран в виде готового образа, который мы можем получить и использовать:

```
docker volume create portainer_data
sudo docker run -d -p 8000:8000 -p 9443:9443
  --name portainer --restart=always -v
  /var/run/docker.sock:/var/run/docker.sock
  -v portainer_data:/data portainer/
  portainer-ce:latest
```

Поскольку такого образа еще нет, система предложит его загрузить, а после загрузки создаст контейнер и запустит его. Обратите внимание: мы будем обращаться к этому контейнеру как к серверу, для чего указываем трансляцию TCP-порта.

После запуска контейнера мы можем запустить на своей машине браузер и посмотреть, что получилось, в графической панели управления докер-контейнерами Portainer по адресу https://localhost:9443 (рис. 4).

При первом обращении нужно задать пароль, после чего стартовая рабочая страница будет выглядеть примерно так, как показано на рис. 5; нас будет интересовать именно сервер local.

Возникает вопрос: кто занимается созданием приложений и другими задачами, связанными с развертыванием и сопровождением подобной инфраструктуры? Ответ на него есть: это **DevOps-инженер** (DevOps — акроним от *англ.* Development & Operations — разработка и информационно-технологическое обслуживание) **— специалист, который совмещает функции разработчика (Dev) и системного администратора (Ops).** Он не разрабатывает само приложение и не занимается, например, настройкой основных серверов и сети, его обязанности ближе к работе системных администраторов, чем программистов. Конечно, эту работу можно поручить непосредственно системным администраторам или программистам, но в большой системе подобных задач будет слишком много, и, скорее всего, программисты будут неправильно настраивать ОС, а системные администраторы не смогут правильно подготовить инструкции для адаптации и сконфигурировать множество сценариев-программ, хотя Docker будет небольшой (пусть и важной) частью реальной инфраструктуры и работы администраторов.

Мы подготовили инфраструктуру, теперь начнем обрабатывать данные. Для примера мы, как указывали выше, воспользуемся данными о рынке недвижимости Москвы, которые собраны и опубликованы Сбербанком[1].

Казалось бы, мы уже можем начать исследовать данные, но… Мы получили данные в формате CSV, а наш инструмент анализа не позволяет работать

[1] Этот набор данных доступен по ссылке: https://github.com/AdmiralWen/Sberbank-Russian-Housing-Market/tree/master/Data

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

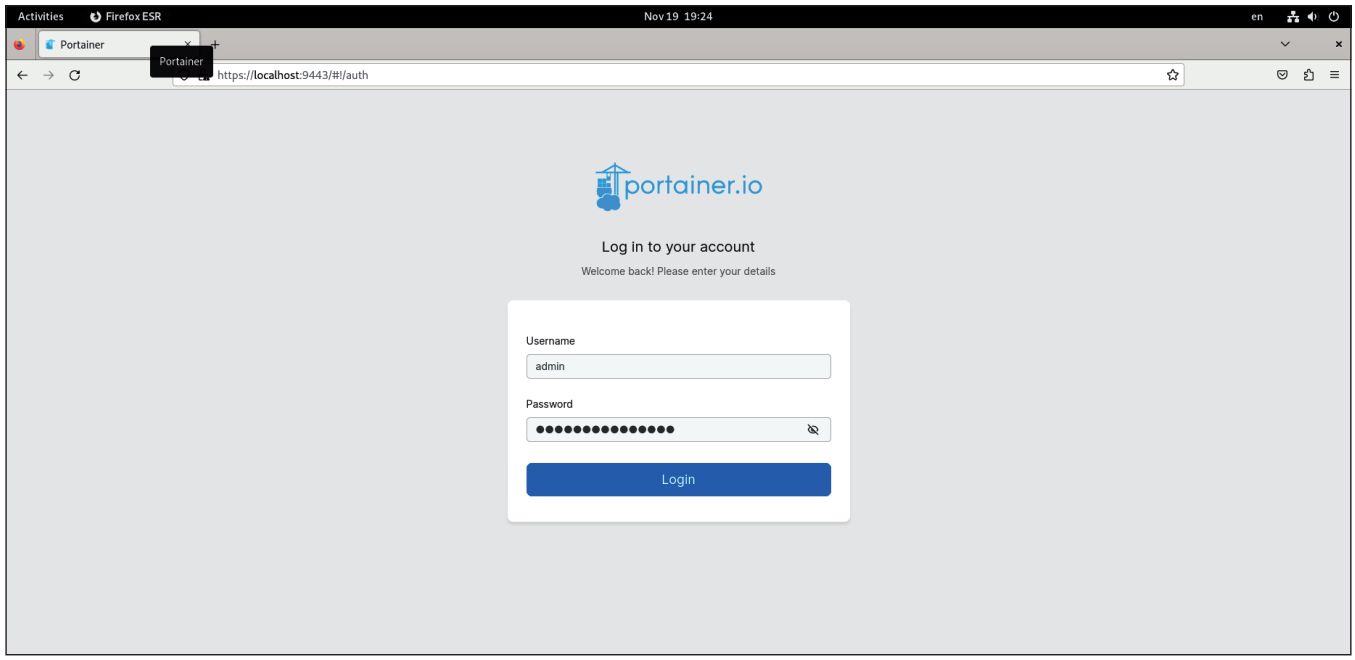

Рис. 4. Вход в веб-приложение для управления средой Docker
Fig. 4. Login to web-application for managing the Docker environment

с таким источником. Кроме того, мы понимаем, что количество источников (и самых разных проблем с ними) будет постоянно расти. К обязанностям системных администраторов и инженеров-DevOps решение этих задач не относится, а программисты смогут начать работать только тогда, когда будет определено, что именно надо делать. Кто же построит собственно цепочку обработки данных с учетом всех входящих условий и необходимости дальнейшего развития?

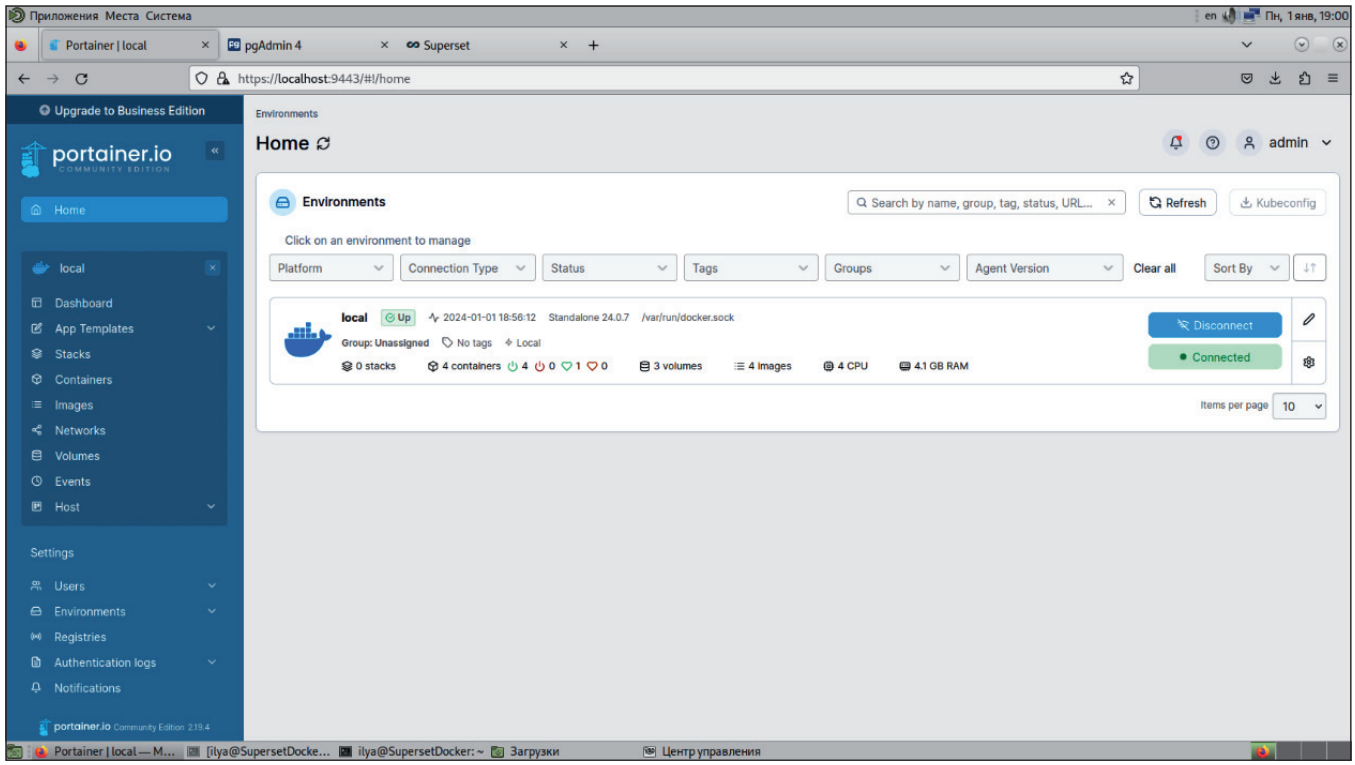

Рис. 5. Входная страница приложения Portainer
Fig. 5. Input page of the Portainer application

ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2024 • Том 39 № 5

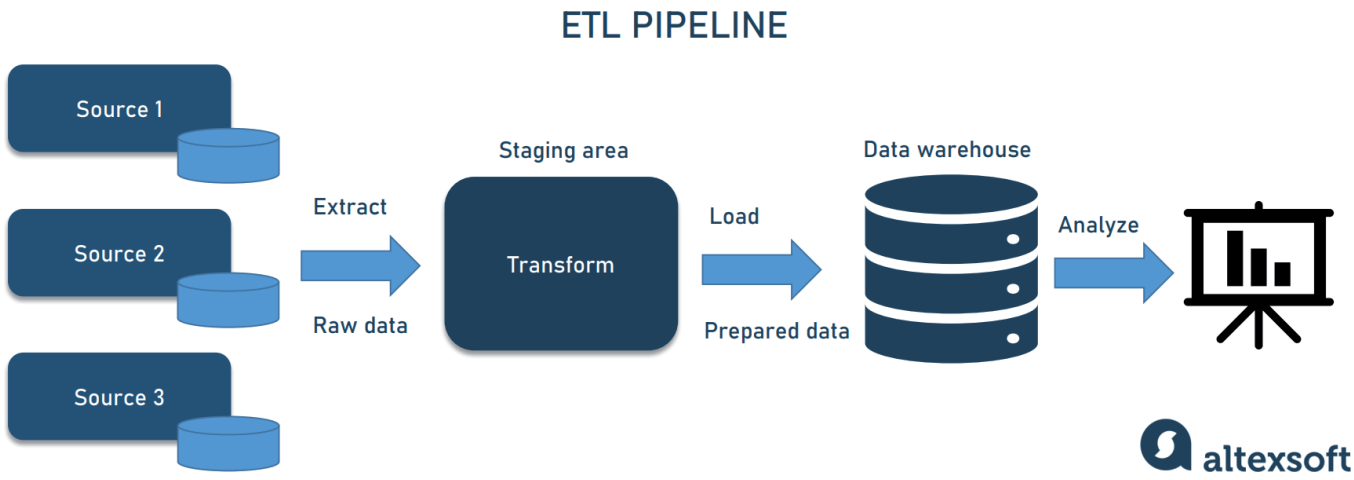

Рис. 6. ETL-конвейер[1]
Fig. 6. ETL pipeline

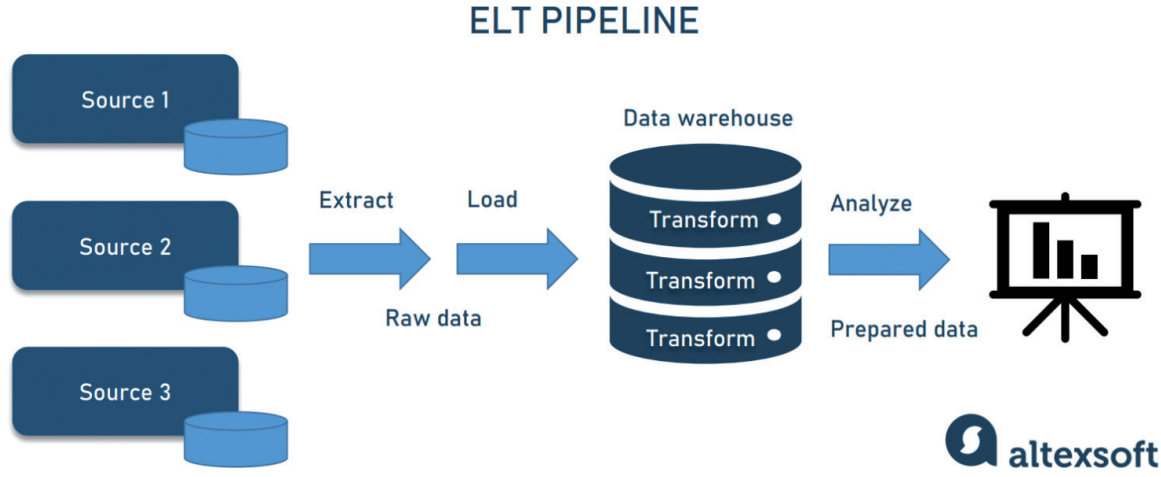

Рис. 7. ELT-конвейер[2]
Fig. 7. ELT pipeline

Решением таких задач занимаются также отдельные специалисты — **инженеры данных (*англ.* Data Engineer), которые организуют сбор информации, ее хранение и первоначальную обработку**[3]. Поскольку мы знаем, что в нашем случае данные будут поступать постоянно, то инженер не станет анализировать каждую ошибку в отдельности (постоянно наступая на новые грабли). Необходимо спланировать процесс их поступления, хранения и предварительной обработки так, чтобы решать потом разные возникающие задачи — задачи бизнес-аналитики, Data Science, машинного обучения. Методы, применяемые для этого, могут быть самыми разными [13].

Инженер обеспечивает представление данных в виде потока, который проходит три основных этапа:

1. *Потребление данных* (*англ.* Data Ingestion) — поступление данных из множества различных источников в целевую систему. Они могут быть и структурированными (например, данные о квартирах), и неструктурированными (например, статьи о влиянии экономической ситуации на текущие цены).
2. *Преобразование данных* (*англ.* Data Transformation) — преобразование данных под задачи наших конечных пользователей, а также устранение ошибок, преобразование данных в единые форматы, устранение дубликатов.
3. *Обслуживание данных* (*англ.* Data Serving), или подача данных, — предоставление данных в виде единого пульта (*англ.* Dashboard), или

[1] Источник изображения см.: Data Engineering: Concepts, Processes, and Tool. *AltexSoft*. 13.03.2023. https://www.altexsoft.com/blog/what-is-data-engineering-explaining-data-pipeline-data-warehouse-and-data-engineer-role/

[2] Источник изображения см.: там же.

[3] Подробнее об этой специальности см.: Data Engineering: Concepts, Processes, and Tools. *AltexSoft*. 13.03.2023. https://www.altexsoft.com/blog/what-is-data-engineering-explaining-data-pipeline-data-warehouse-and-data-engineer-role/ или перевод этой статьи: Data Engineering: концепции, процессы и инструменты. *Хабр*. 24.07.2023. https://habr.com/ru/articles/743308/

источника данных, конечному потребителю, команде аналитиков (дата-сайентистам), на платформу бизнес-аналитики и т. д.

Для создания системы сбора данных инженер данных строит конвейер их обработки и, скорее всего, использует два основных подхода.

Первый, наиболее известный, — это ETL-конвейер (*англ.* Extract, Transform, Load — извлечение, преобразование, загрузка) (рис. 6). Он автоматизирует следующие процессы:

- *Extract* — извлечение данных. В начале конвейера мы получаем «сырые» данные из множества источников: из отдельных файлов, баз данных, от других приложений и т. д.;
- *Transform* — стандартизация данных. После извлечения данных инженер должен подготовить сценарии, которые преобразуют их, приводя в соответствие с требованиями к форматам (например, к единой форме и шкале времени);
- *Load* — сохранение данных в новое место. После преобразования данных в применимый формат их можно загрузить в целевое хранилище.

ETL-конвейер строят для работы со структурированными данными, используя их в бизнес-аналитике и для представления в виде пультов. А это как раз наш случай.

Второй вид — ELT-конвейер. В нем те же процессы выполняются в другом порядке, что позволяет сохранить исходные данные и позднее решать другую задачу или несколько задач, требующих разных трансформаций (рис. 7).

Поскольку мы используем структурированные данные и хотим их визуализировать, **будем применять ETL-конвейер (его упрощенную версию), опираясь на свободно-распространяемые инструменты:**

- использовать данные из заранее найденных файлов CSV (на самом деле эти файлы представляют собой результат работы такой системы);
- хранить и предварительно обрабатывать данные с помощью СУБД PostgreSQL; преобразовывать и анализировать их мы будем только для примера, чтобы показать, с какими проблемами можем столкнуться;
- подавать данные конечному потребителю с помощью платформы Apache Superset, чтобы увидеть, что именно такие платформы делают.

Для развертывания используем готовую инфраструктуру. Скорее всего, ***ставить и разворачивать образы будет специалист DevOps,*** потому что потом администрировать получившийся комплекс вручную будет очень трудно, а «перетаскивать» данные через конвейер вручную можно только во время тестов и проб. Тем не менее ***формулировать задачу инженеру DevOps будет инженер данных*** (скорее всего, опытный специалист на должности архитектора данных).

Рассмотрим далее работу инженера данных.

1-й этап. Получение данных: запустим образ инструмента анализа данных и их визуализации — Apache Superset (рис. 8).

Мы можем воспользоваться специально установленным нами приложением для управления

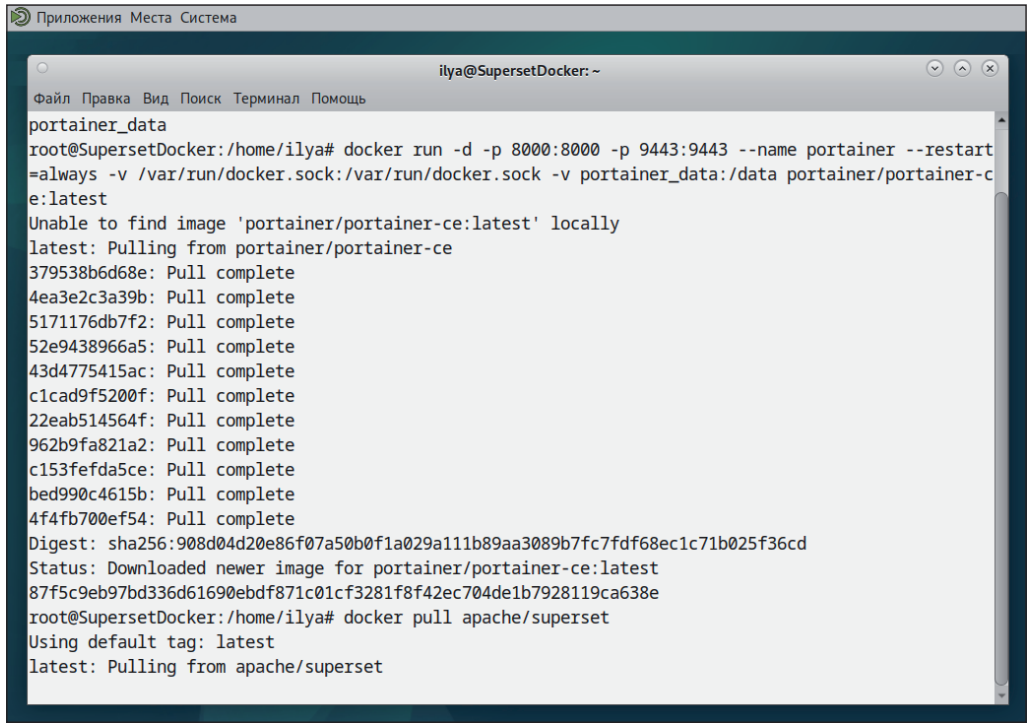

Рис. 8. Процесс скачивания образов и подготовки контейнеров среды Docker в командном интерпретаторе
Fig. 8. The process of downloading images and preparing Docker environment containers in the command interpreter

ISSN 0234-0453 • ИНФОРМАТИКА И ОБРАЗОВАНИЕ • 2024 • Том 39 № 5

(Portainer), а можем выполнять все настройки через командную строку.

Покажем установку и запуск в виде команд, поскольку это более компактный способ:

```
docker pull apache/superset
docker run -d -p 8080:8088 -e
  «SUPERSET_SECRET_KEY=strong_secret_key_here»
  --name superset apache/superset
```

Получить образ и запустить контейнер на его основе, к сожалению, недостаточно: нужно обновить структуру данных внутри контейнера на основе свежих изменений продукта, получить с сайта разработчика свежие примеры, инициализировать — и уже после этого обращаться к работающему приложению.

```
docker exec -it superset superset db upgrade
docker exec -it superset superset load_examples
docker exec -it superset superset init~
docker exec -it superset superset init
docker exec -it superset superset fab
  create-admin --username admin --password admin
  --firstname Superset --lastname Admin --email
  admin@superset.com
```

Важно отметить, что в качестве параметра нужно указать имя и пароль для управления контейнером. Если вы их забыли или они не заработали — есть и команда для их сброса:

```
docker exec -it superset superset fab
  reset-password --username admin --password
  admin
```

Кроме системы отображения, нам потребуется хранилище. Поскольку у нас есть готовая инфраструктура, мы можем скачать и установить два готовых продукта: собственно сервер БД PostgreSQL и графическое средство управления pgAdmin:

```
docker pull postgres
docker run --name some-postgres -e
  POSTGRES_PASSWORD=admin -d postgres
docker pull dpage/pgadmin4
docker run -p 8081:80 -e
  'PGADMIN_DEFAULT_EMAIL=user@domain.com' -e
  'PGADMIN_DEFAULT_PASSWORD=admin' -e
  'PGADMIN_CONFIG_ENHANCED_COOKIE_PROTECTION=
  True' -e
  'PGADMIN_CONFIG_LOGIN_BANNER=»Authorised
  users only!»' -e
  'PGADMIN_CONFIG_CONSOLE_LOG_LEVEL=10' -d
  dpage/pgadmin4
```

В итоге запущенные у нас контейнеры будут выглядеть в системе управления так, как это представлено на рисунке 9. Обратим внимание, что СУБД мы не публикуем, потому что обращаться к ней извне не планируем.

2-й этап. Создание БД для работы.

Обратимся к установленной системе управления pgAdmin (http://localhost:8081). Зарегистрируем сервер БД (рис. 10).

Укажем необходимые параметры соединения (рис. 11).

IP-адрес сервера мы видели на снимке запущенных контейнеров. Имя пользователя у нас одно —

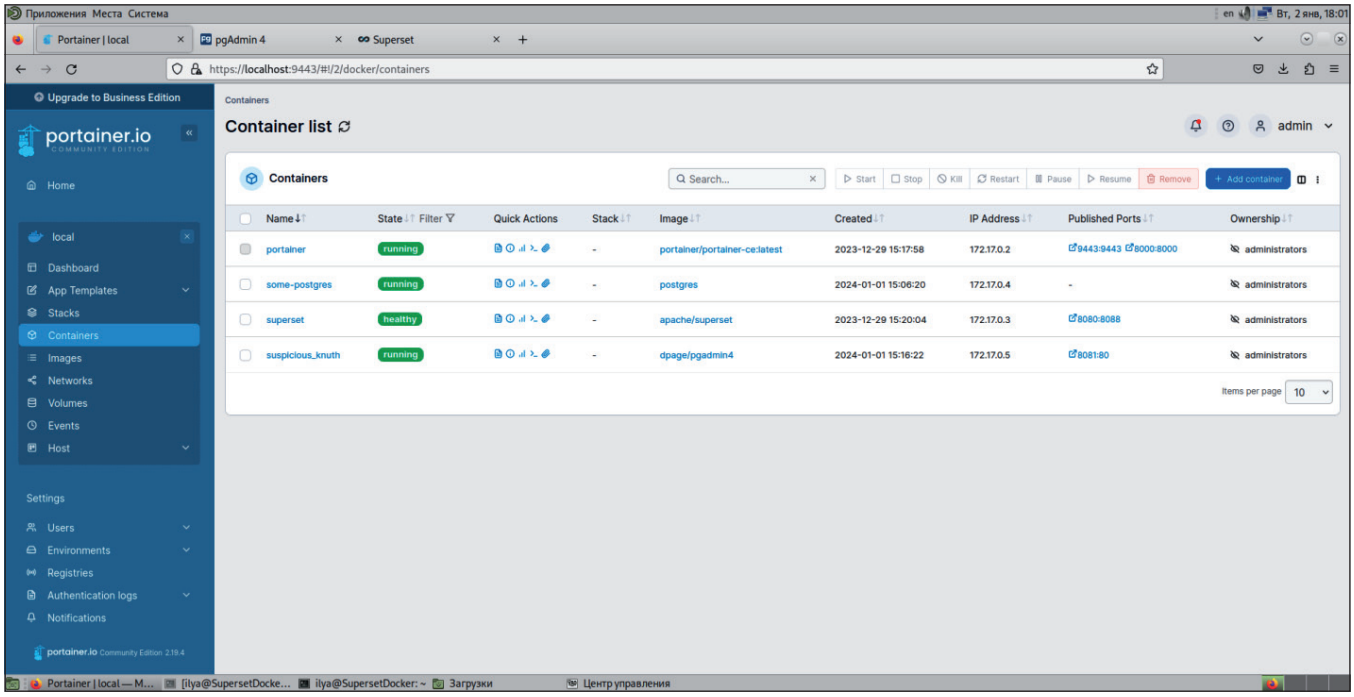

Рис. 9. Список работающих контейнеров Docker

Fig. 9. List of running Docker containers

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

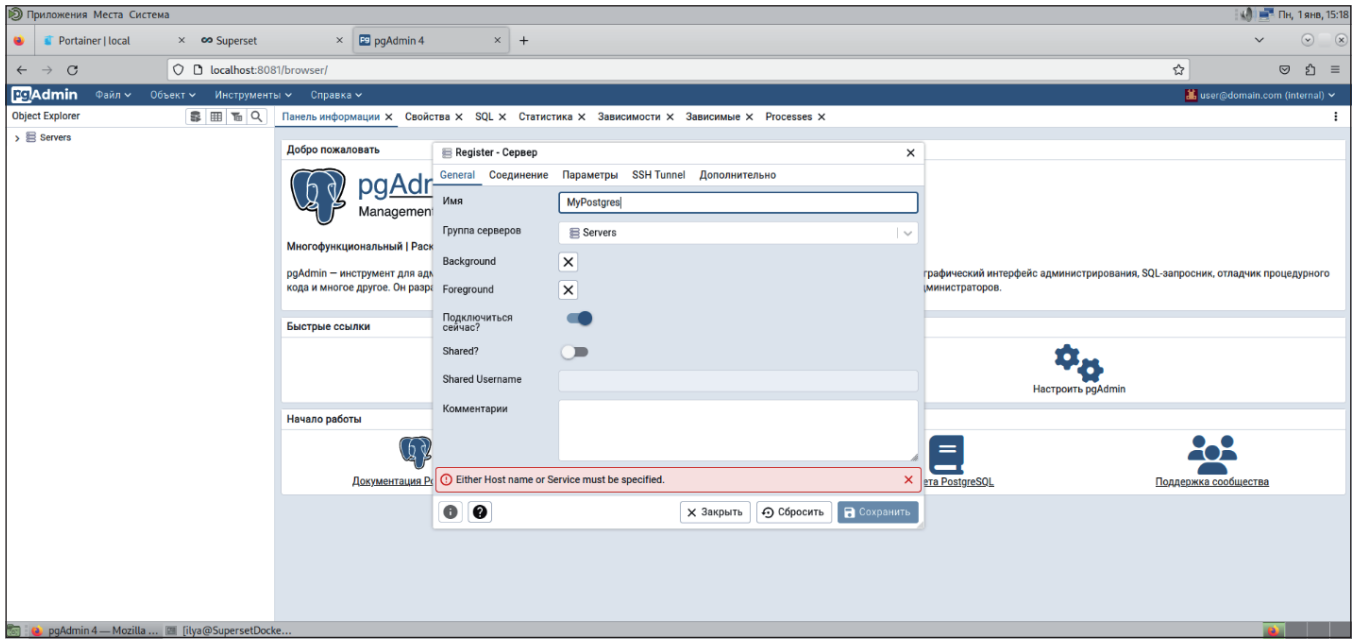

Рис. 10. Регистрация сервера базы данных в приложении управления pgAdmin
Fig. 10. Database server registration in the pgAdmin management application

postgres (это администратор, и в реальной ситуации нам было бы категорически запрещено использовать его при работе с отдельными БД по требованиям информационной безопасности), а пароль мы задали сами — в строке запуска контейнера: admin.

Подключившись к серверу, мы можем создать рабочую БД (рис. 12). Назовем ее SberRealty. Теперь все готово для загрузки данных и их визуализации.

3-й этап. Загрузка данных и их визуализация.

Используем набор данных, предоставленный Сбербанком для публичного конкурса на создание средств прогнозирования продажи недвижимости в 2017 году.

Основные поля в этом файле:

- price_doc: цена продажи;
- id: идентификатор;

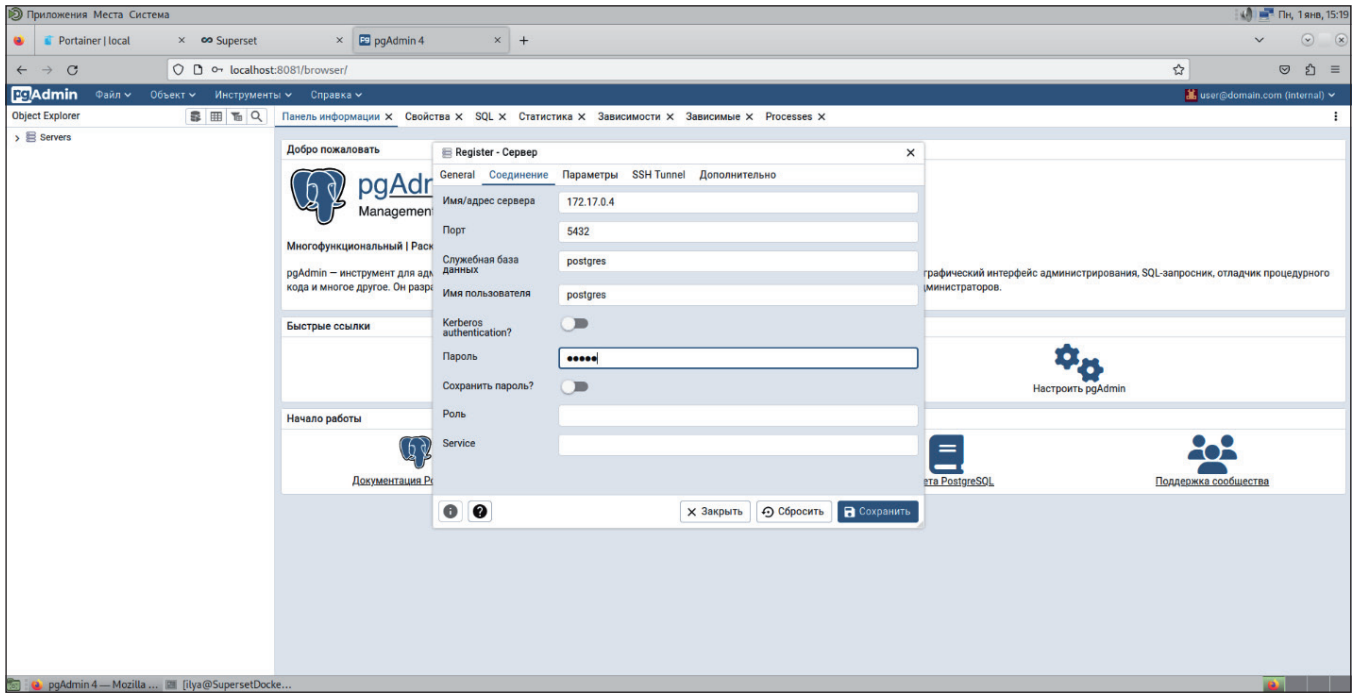

Рис. 11. Параметры соединения с сервером в приложении pgAdmin
Fig. 11. Server connection parameters in the pgAdmin application

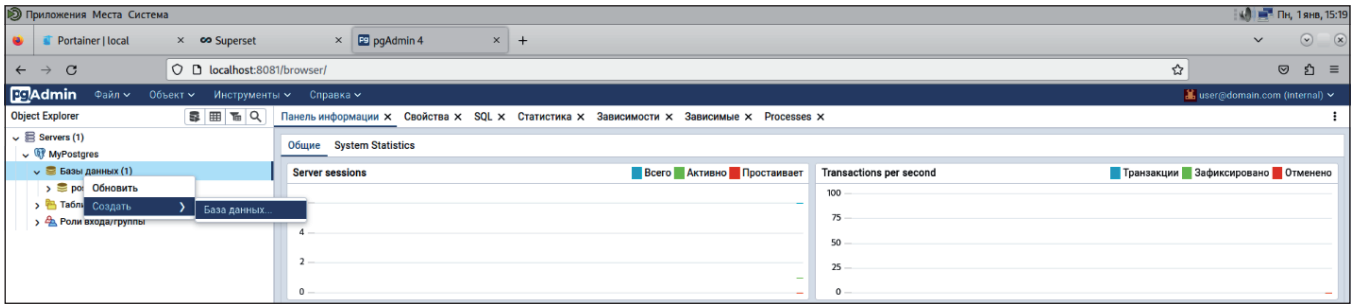

Рис. 12. Создание базы данных
Fig. 12. Creating a database

- timestamp: дата транзакции;
- full_sq: полная площадь помещения;
- life_sq: жилая площадь;
- floor: этаж;
- max_floor: количество этажей в здании;
- material: материал стен;
- build_year: год постройки;
- num_room: количество жилых комнат;
- kitch_sq: площадь кухни;
- state: состояние квартиры;
- product_type: тип покупки — для проживания или для инвестиций;
- sub_area: район.

Остальные поля (а всего их в файле 242) — это описание общей макроэкономической ситуации на момент совершения сделки.

В приложении Apache Superset создаем подключение к БД. При этом приложение сообщает, что для подключения нужно установить средство работы с БД, и предлагает выбрать одно из четырех возможных. Выбираем PostgreSQL (рис. 13).

Далее указываем параметры для подключения (рис. 14).

Если все сделано правильно, то, нажав кнопку Connect, мы подключимся к БД и у нас появится новая БД в списке (рис. 15).

Важная техническая деталь: изначально у нас не будет возможности загружать в БД файлы с данными. Нужно отредактировать свойства БД: на закладке Advanced в разделе Security поставить «галочку» у «Allow file uploads to database», как это показано на рисунке 16 (в документации процедуру отразили неправильно).

Теперь загружаем данные в БД: выбираем пункт Upload CSV в верхнем меню (рис. 17).

Выбираем нужный файл и выполняем загрузку (рис. 18).

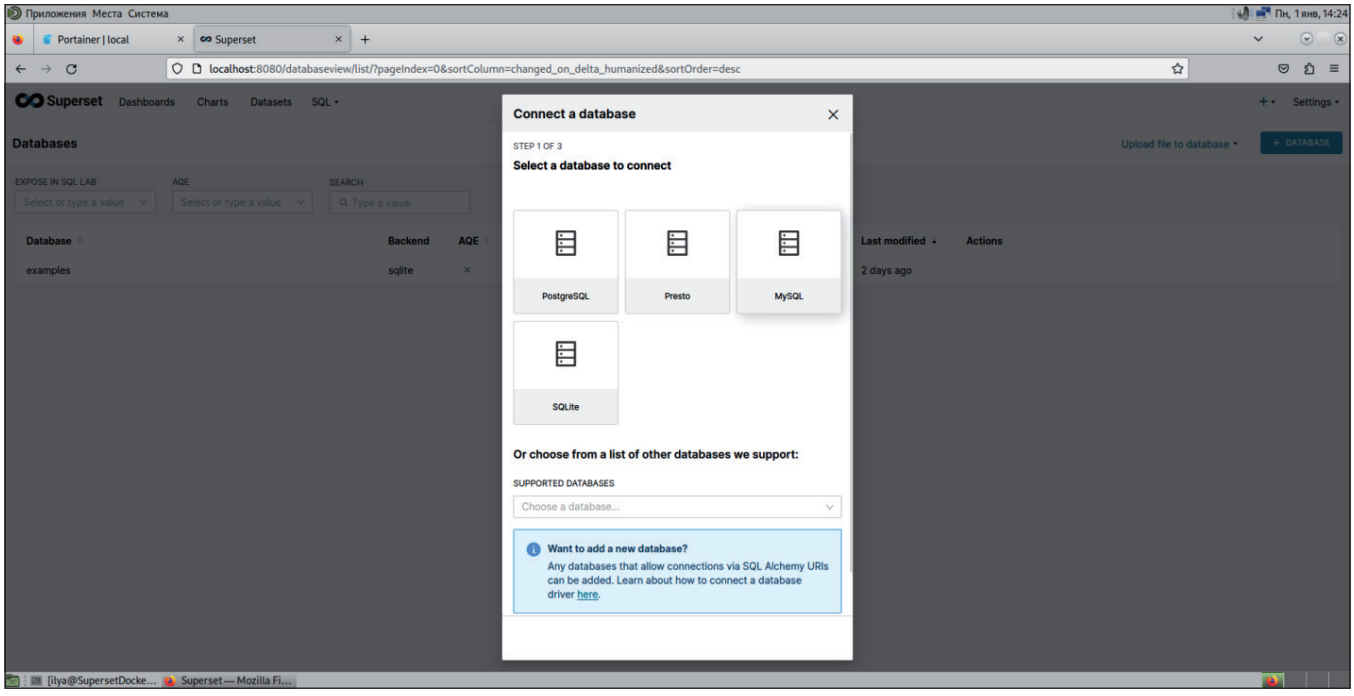

Рис. 13. Подключение к базе данных в приложении Apache Superset: выбор средства работы с базами даных
Fig. 13. Connecting to the database in the Apache Superset application: selecting a database tool

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

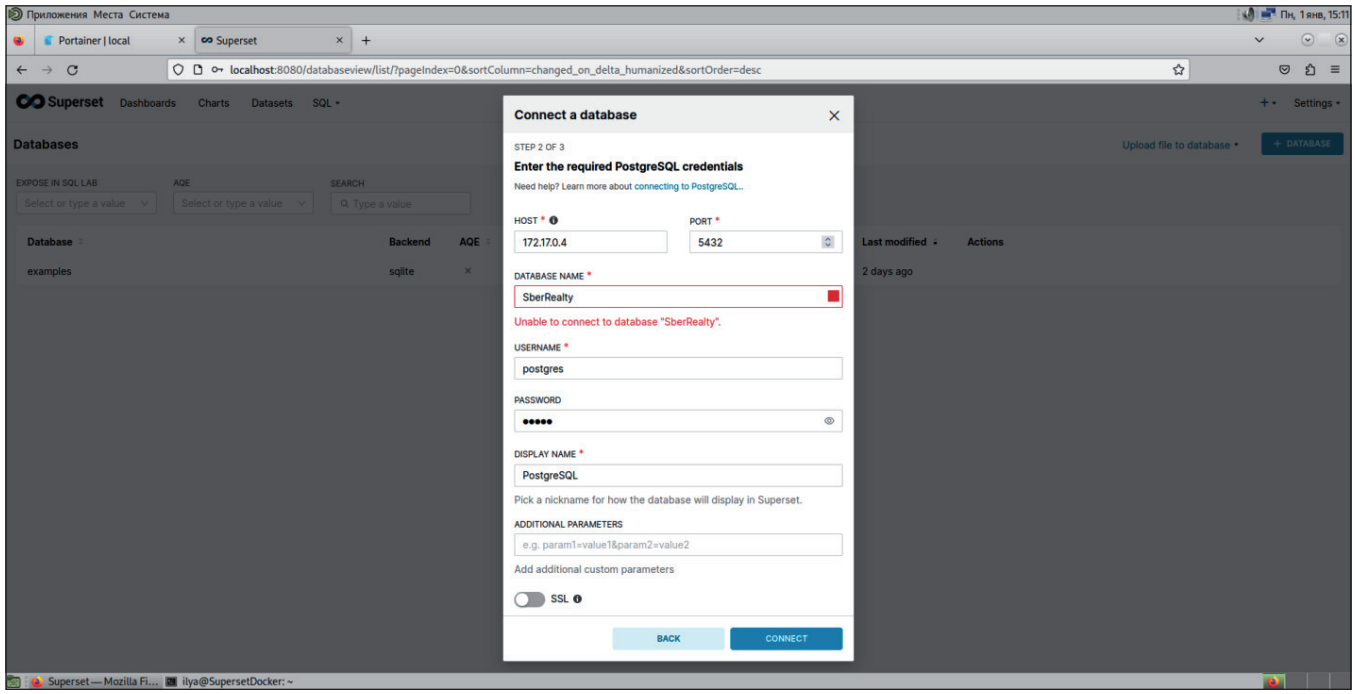

Рис. 14. Подключение к базе данных в приложении Apache Superset: параметры подключения
Fig. 14. Connecting to the database in the Apache Superset application: connection parameters

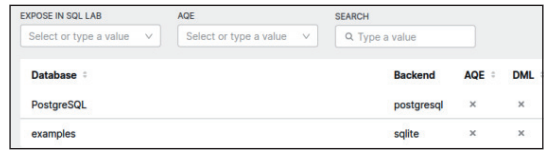

Рис. 15. Результаты подключения базы данных
Fig. 15. Results of database connection

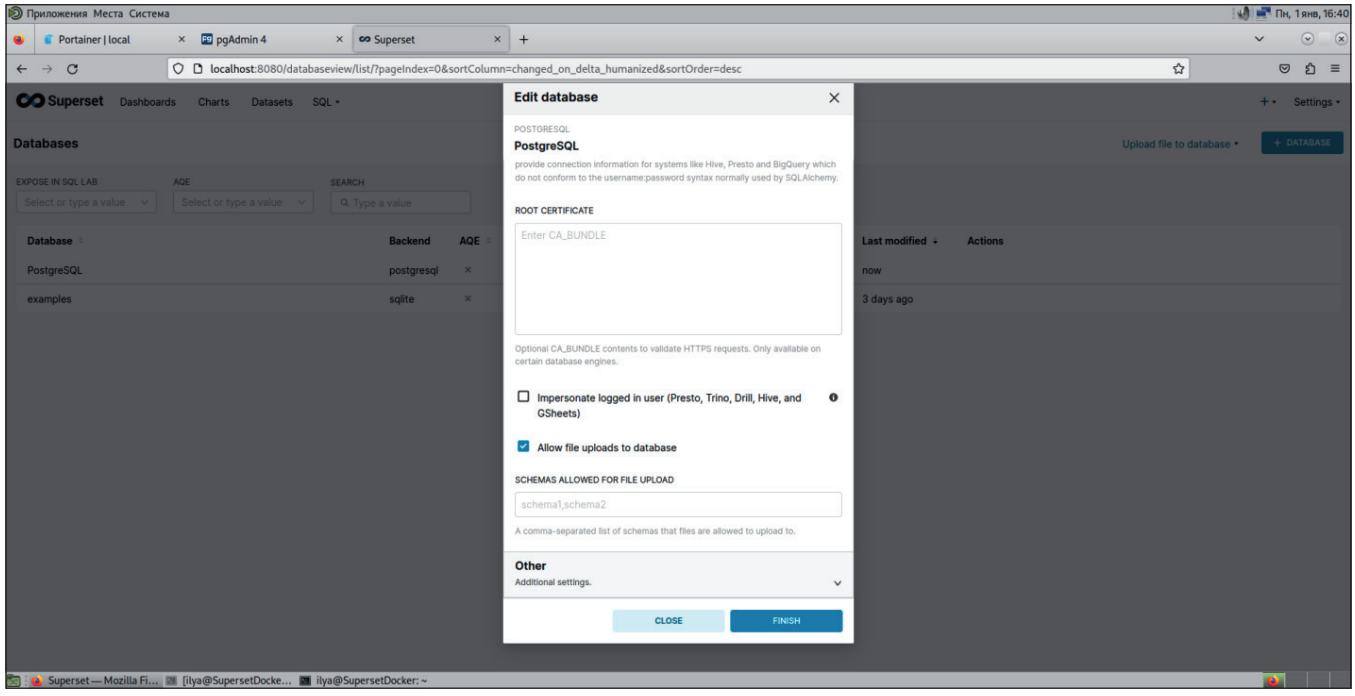

Рис. 16. Разрешаем загрузку данных в базу из файлов
Fig. 16. Allow uploading data to the database from files

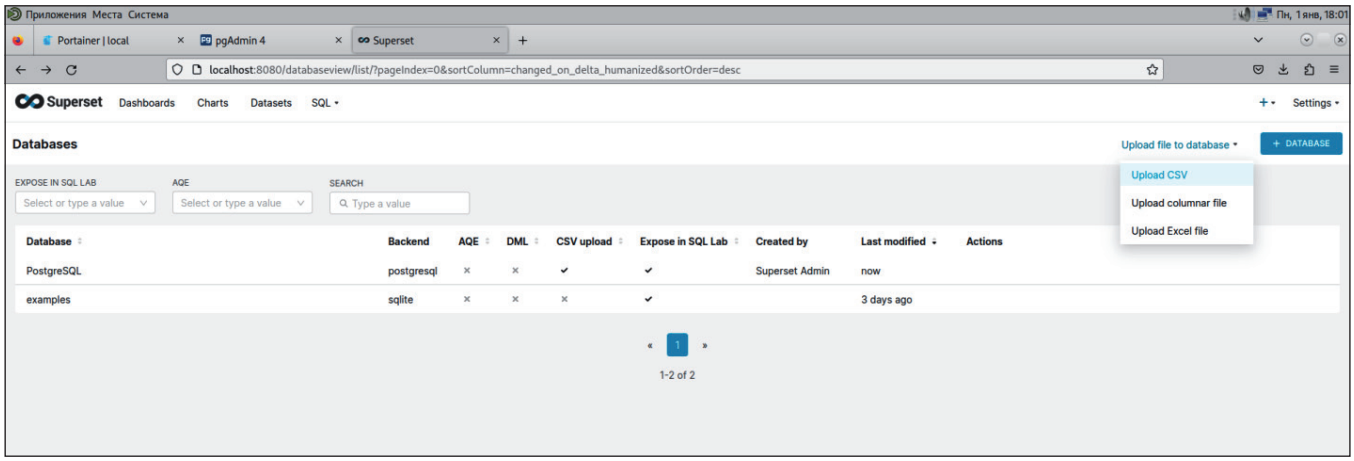

Рис. 17. Начало загрузки данных из файла CSV
Fig. 17. Start loading data from CSV file

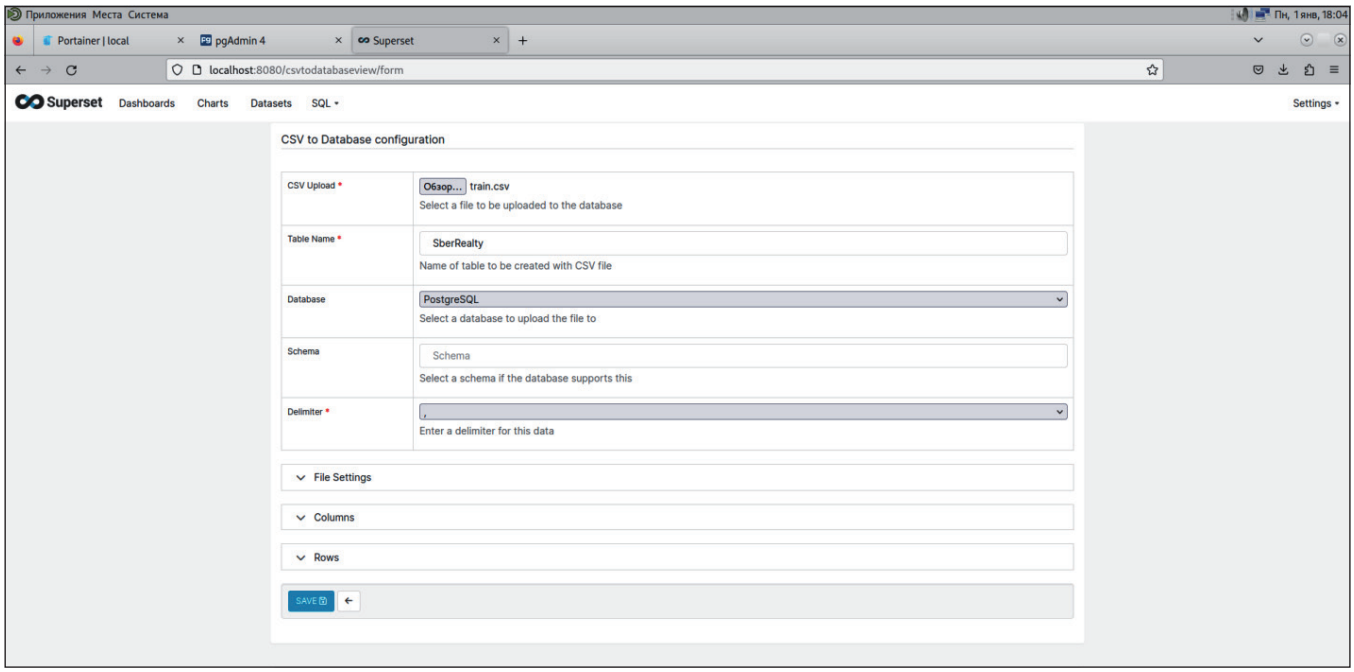

Рис. 18. Параметры загрузки данных из файла CSV в базу данных
Fig. 18. Parameters of uploading data from CSV file to the database

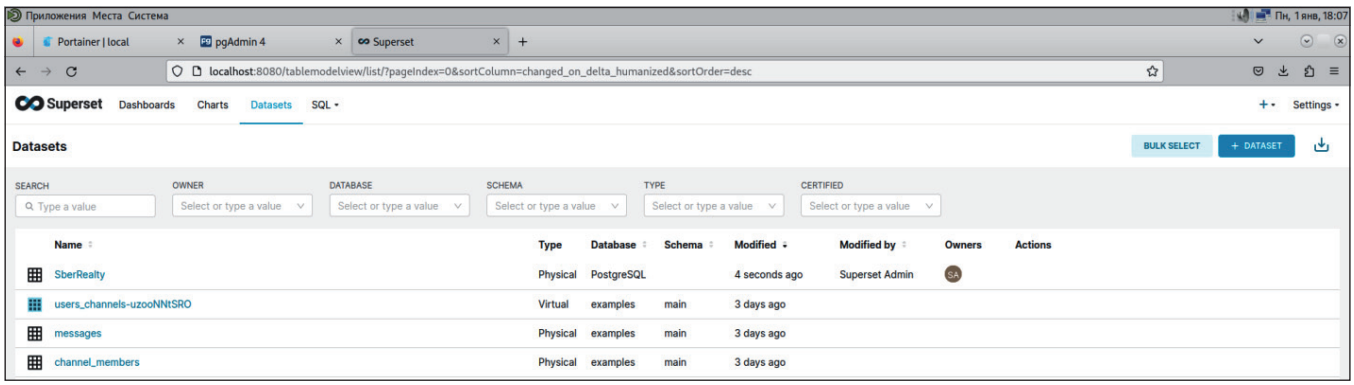

Рис. 19. Новый набор данных после загрузки данных
Fig. 19. New dataset after data uploading

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

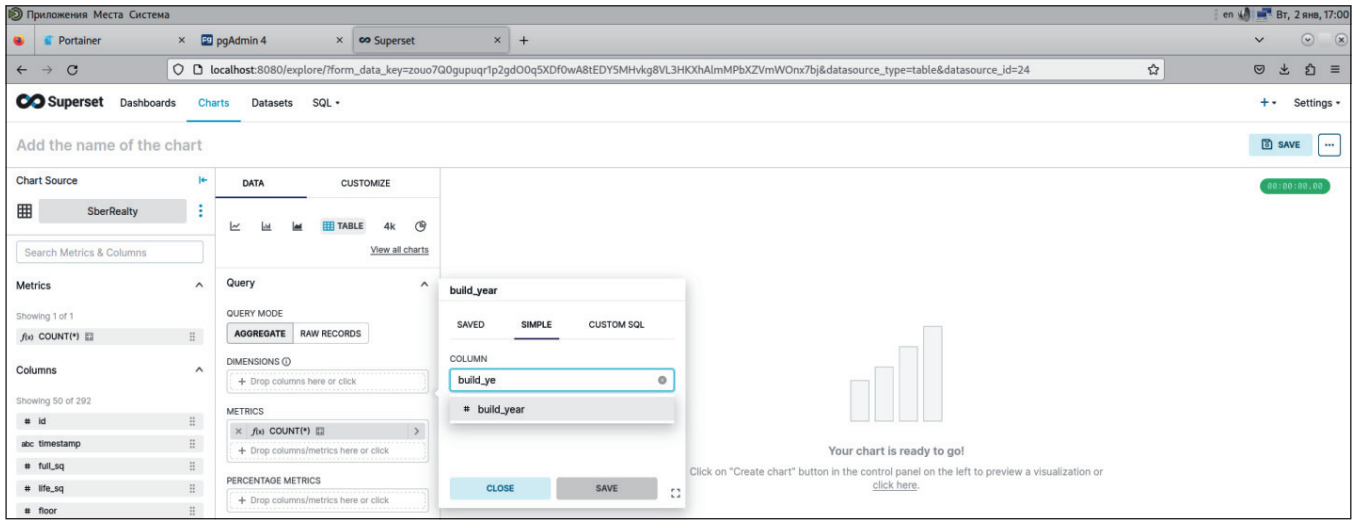

Рис. 20. Создание диаграммы: выбор поля
Fig. 20. Creating a diagram: selecting a field

Появился новый набор данных (Dataset). Мы можем его увидеть в списке доступных наборов данных (рис. 19).

Если мы откроем наш новый набор данных SberRealty, то сервис предложит создать диаграмму для его визуализации. Визуализация основывается на запросе к набору данных (в нашем случае это уже запрос агрегации — Agregate) и способе визуализации (столбчатая диаграмма). Первое, что нужно сделать, — это выбрать поле для анализа: щелкнуть курсором в пункте Dimension и при выборке по образцу найти поле build_year (поскольку всего полей 242, то лучше начать набирать название). После выбора нажмем кнопку Save (рис. 20).

В разделе My metric выберем единственную уже сохраненную функцию COUNT(*) (как известно, эта функция подсчитывает количество) и нажмем Save (рис. 21).

Первая диаграмма для этого набора данных представлена на рисунке 22.

Нужно отметить, что операция Transform, конечно, не показала ошибок, но данные нуждаются в анализе и очистке. 13,6 тысяч строк вообще не имеют данных о годе постройки, у 530 строк год постройки обозначен как «0», а 368 строк имеют год постройки «1».

Сохраним созданную диаграмму и создадим первое табло (пульт), как это показано на рисунке 23. Наберем название и нажмем «Save & go to dashboard».

Как выглядит готовый пульт в приложении Apache Superset, показано на рисунке 24.

Заметим, что из таких элементов можно набрать сложные и очень информативные комплексы. Возможности Apache Superset демонстрируют многочисленные примеры пультов и диаграмм, которые сразу предоставляются вместе с продуктом (рис. 25).

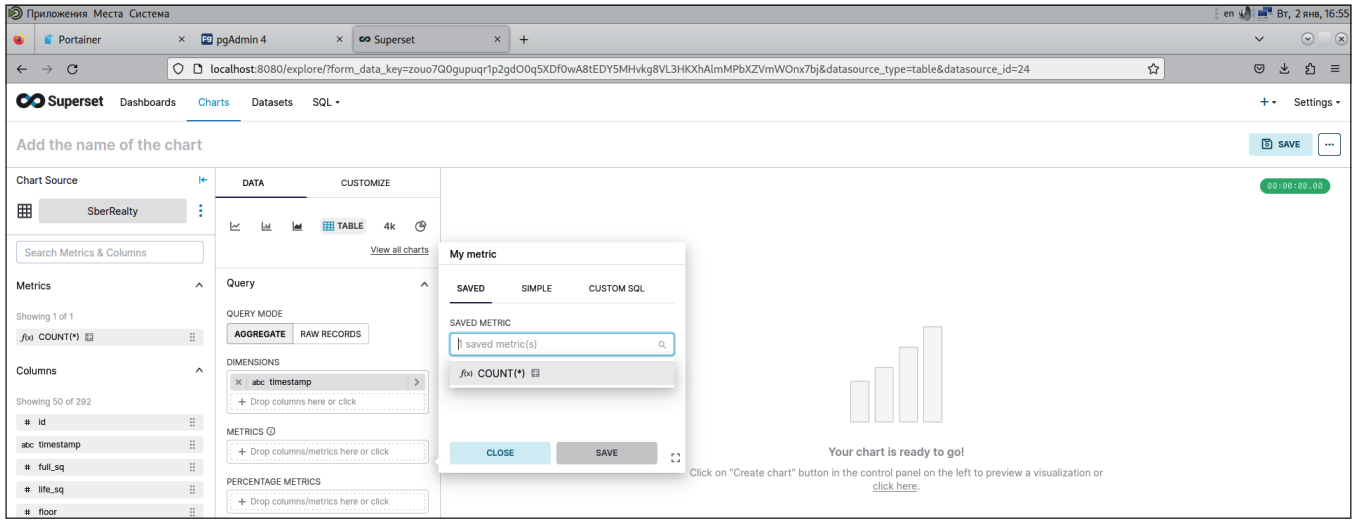

Рис. 21. Создание диаграммы: выбор способа агрегации
Fig. 21. Creating a diagram: selecting the aggregation method

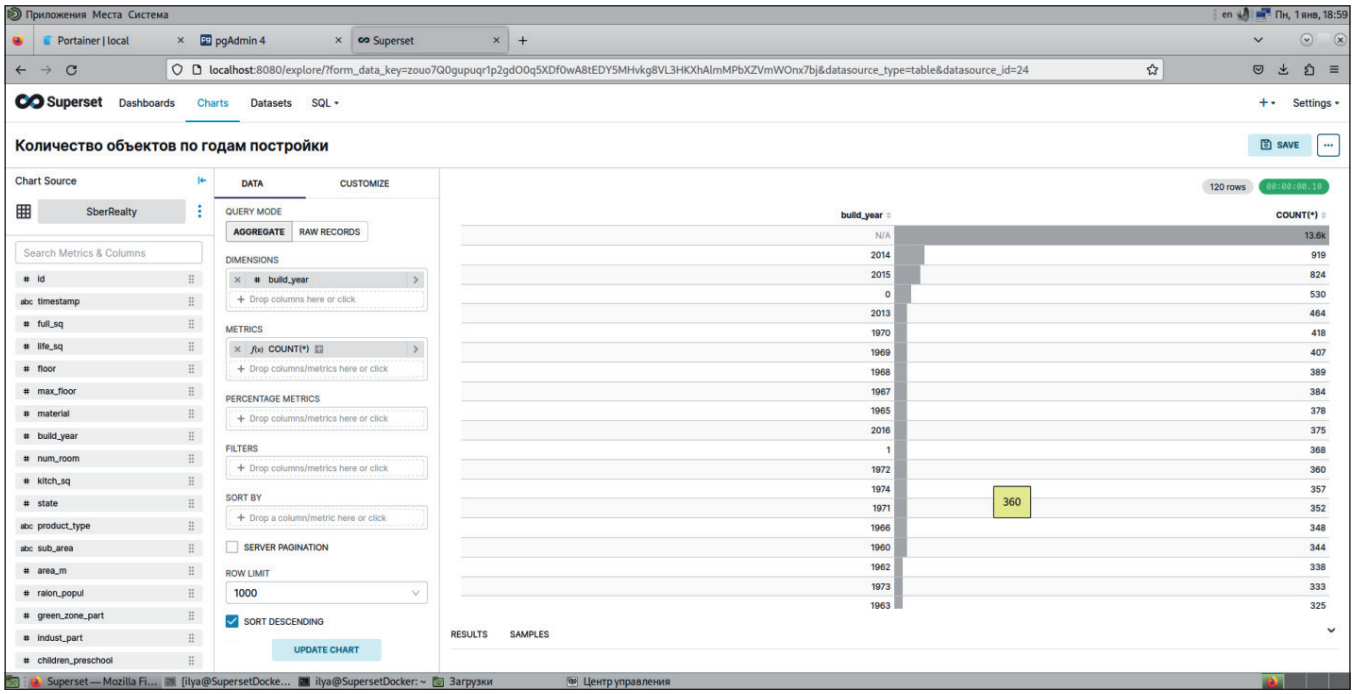

Рис. 22. Новая диаграмма, общий вид
Fig. 22. New diagram, general view

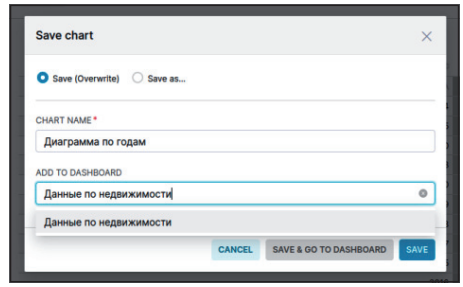

Рис. 23. Сохранение диаграммы и создание нового пульта
Fig. 23. Saving the diagram and creating a new dashboard

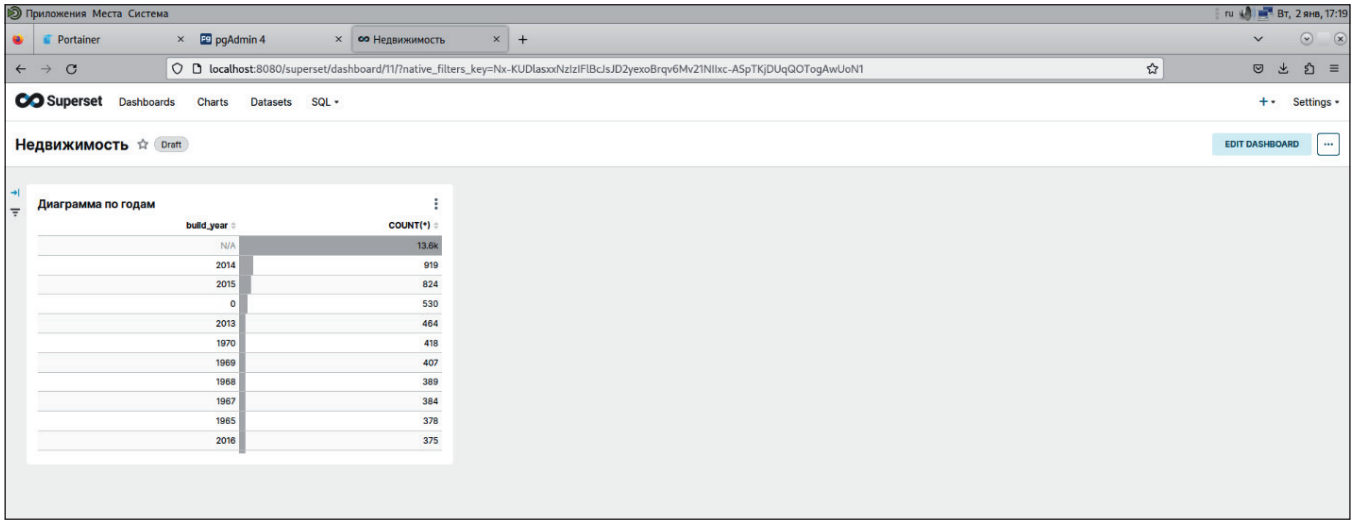

Рис. 24. Пульт в приложении Apache Superset
Fig. 24. Dashboard in Apache Superset application

ISSN 0234-0453 • INFORMATICS AND EDUCATION • 2024 • Volume 39 No 5

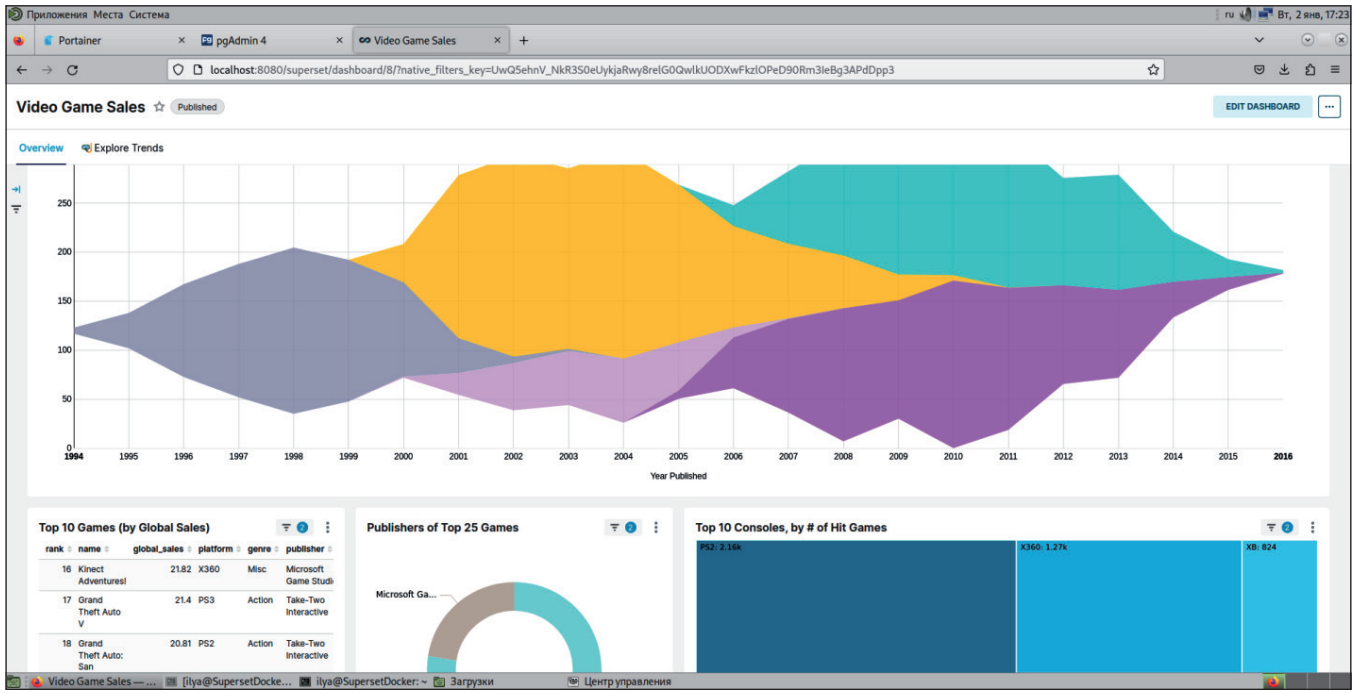

Рис. 25. Пример готового пульта — набора блоков
Fig. 25. Example of a ready dashboard — a set of blocks

5. Заключение

В приведенном примере рассмотрен далеко не полный спектр проблем и задач, который решают инженеры данных и DevOps-инженеры. В реальных условиях им пришлось бы заранее предусматривать автоматические средства фильтрации и очистки данных, заранее проектировать структуру, позволяющую не только сохранить весь поток больших данных, но и визуализировать его, причем и в моменте (для оперативного мониторинга), и в виде агрегированных данных, необходимых для принятия решений. В реальности это может оказаться очень непросто[1]. В дальнейшем данные в рамках подготовленной инфраструктуры могут быть использованы для самых разных целей. Например, именно такой набор данных чаще всего нужен для построения различного рода рекомендательных систем [17, 18].

Из приведенного очень простого примера видно, сколько действий пришлось предпринять, чтобы создать даже такой упрощенный конвейер обработки и анализа данных. И на этом работа с данными не остановится, ими займутся уже другие специалисты — менеджеры при принятии решений, аналитики данных при поиске и иллюстрировании найденных закономерностей, а также другие специалисты предметных областей (от обеспечивающих снабжение до технологов производства) [19, 20].

Отметим также, что нет четких границ, которые позволят отделить задачи DevOps-инженера от задач системного администратора или должностные обязанности инженера-разработчика программного обеспечения от должностных обязанностей инженера данных. Их навыки и знания во многом пересекаются, более того — им необходимо знать пусть и не всю, но основную специфику работы смежников. На небольшом или типовом проекте специалисты по информационным технологиям могут их совмещать. Но только полная команда, включающая в себя системных администраторов, специалистов по телекоммуникационным сетям, аналитиков Data Science и т. д., справится с организацией работы с большими данными.

Финансирование
Работа выполнена в рамках государственного задания Министерства просвещения Российской Федерации (тема № 124052100092-0 «Вариативное обучение основам искусственного интеллекта в общем образовании на основе интегративного подхода»).

Funding
The research was carried out within the state assignment of the Ministry of Education of the Russian Federation (theme No 124052100092-0) “Variative teaching of the basics of artificial intelligence in general education based on the integrative approach”).

Список источников / References

1. *Самылкина Н. Н.* Информатика, углубленный уровень. Реализация требований ФГОС среднего общего образования: методическое пособие для учителя. М.: Институт стратегии развития образования; 2023. 226 с. EDN: LGYRRI.

[1] *Митрофанов М.* Тот самый датасет, где архитектор чуть не сошел с ума. *Хабр.* 16.11.2023. https://habr.com/ru/companies/omk-it/articles/774126/

[*Samylkina N. N.* Informatics, advanced level. Realization of the requirements of the FSES of secondary general education: A methodological manual for teachers. Moscow, Institute for Strategy of Education Development; 2023. 226 p. (In Russian.) EDN: LGYRRI.]

2. *Lee S. J., Kwon K.* A systematic review of AI education in K-12 classrooms from 2018 to 2023: Topics, strategies, and learning outcomes. *Computers and Education: Artificial Intelligence*. 2024;6:100211. DOI: 10.1016/j.caeai.2024.100211.

3. *Israel-Fishelson R., Moon P. F., Tabak R., Weintrop D.* Preparing students to meet their data: An evaluation of K-12 data science tools. *Behaviour & Information Technology*. 2023:1–20. DOI: 10.1080/0144929X.2023.2295956.

4. *Hazzan O., Ragonis N., Lapidot T.* Data science and computer science education. *Guide to Teaching Computer Science*. Springer, Cham; 2020:95–117. DOI: 10.1007/978-3-030-39360-1_6.

5. Foundations of data science for students in grades K-12: Proceedings of a workshop. Washington, DC, The National Academies Press; 2023. 152 p. DOI: 10.17226/26852.

6. *Ткач Т. В.* Машинное обучение и обработка больших данных в условиях современной школы. *Информатика в школе*. 2020;(7(160)):25–29. EDN: JETIPH. DOI: 10.32517/2221-1993-2020-19-7-25-29.

[*Tkach T. V.* Machine learning and Big Data processing in a modern school. *Informatics in School*. 2020;(7(160)):25–29. (In Russian.) EDN: JETIPH. DOI: 10.32517/2221-1993-2020-19-7-25-29.]

7. *Самылкина Н. Н., Салахова А. А.* Обучение основам искусственного интеллекта и анализа данных в курсе информатики на уровне среднего общего образования: монография. М.: МПГУ; 2022. 242 с. EDN: BACMCW. DOI: 10.31862/9785426310643.

[*Samylkina N. N., Salakhova A. A.* Teaching the basics of artificial intelligence and data analysis in the informatics course at the secondary general education level: Monograph. Moscow, MPSU; 2022. 242 p. (In Russian.) EDN: BACMCW. DOI: 10.31862/9785426310643.]

8. *Лейн Д.* Машинное обучение для детей. Практическое введение в искусственный интеллект. М.: Лаборатория знаний; 2023. 288 с.

[*Lane D.* Machine learning for children. A practical introduction to artificial intelligence. Moscow, Laboratoriya Znaniy; 2023. 288 p. (In Russian.)]

9. *Lynch C.* How do your data grow? *Nature*. 2008;455(7209):28–29. DOI: 10.1038/455028a.

10. *Корнев М. С.* История понятия «большие данные» (Big Data): словари, научная и деловая периодика. *Вестник РГГУ. Серия: История. Филология. Культурология. Востоковедение*. 2018;(1(34)):81–85. EDN: YXCBFP. DOI: 10.28995/2073-6355-2018-1-81-85.

[*Kornev M. S.* History of the term “Big Data”. Dictionaries, academic and business periodicals. *Vestnik RGGU. Seriya: Istoriya. Filologiya. Kul’turologiya. Vostokovedeniye*. 2018;(1(34)):81–85. (In Russian.) EDN: YXCBFP. DOI: 10.28995/2073-6355-2018-1-81-85.]

11. *Бессонов В. А.* Какие возможности для статистики цен открывают новые технологии? *Деньги и кредит*. 2021;80(1):120–126. EDN: NCADLU. DOI: 10.31477/rjmf.202101.120.

[*Bessonov V. A.* What opportunities do new technologies bring about for price statistics? *Den’gi i Credit*. 2021;80(1):120–126. (In Russian.) EDN: NCADLU. DOI: 10.31477/rjmf.202101.120.]

12. *Салий В. В., Кухаренко Л. В., Ищенко О. В.* Цифровая трансформация экономики и внедрение хранилищ данных на основе больших данных в инфраструктуру компании. *Вестник Академии знаний*. 2021;44(3):208–214. EDN: FIQRSU. DOI: 10.24412/2304-6139-2021-11240.

[*Saly V. V., Kuharenko L. V., Ischenko O. V.* Digital transformation of the economy and implementation of Big Data storage in company infrastructure. *Vestnik Akademii Znaniy*. 2021;44(3):208–214. (In Russian.) EDN: FIQRSU. DOI: 10.24412/2304-6139-2021-11240.]

13. *Смирнов Д. В., Грушо А. А., Забежайло М. И., Тимонина Е. Е.* Система сбора и анализа информации из различных источников в условиях Big Data. *International Journal of Open Information Technologies*. 2021;(9(4)):64–71. EDN: VNVDNQ.

[*Smirnov D. V., Grusho N. A., Zabezhailo M. I., Timonina E. E.* System for collecting and analyzing information from various sources in Big Data conditions. *International Journal of Open Information Technologies*. 2021;(9(4)):64–71. (In Russian.) EDN: VNVDNQ.]

14. *Менщиков А. А., Перфильев В. Э., Федосенко М. Ю., Фабзиев И. Р.* Основные проблемы использования больших данных в современных информационных системах. *Столыпинский вестник*. 2022;4(1):30. EDN: NDZPQO.

[*Menshchikov A. A., Perfiliev V. E., Fedosenko M. Yu., Fabziev I. R.* The main problems of use of Big Data in modern information systems. *Stolypin Annals*. 2022;4(1):30. (In Russian.) EDN: NDZPQO.]

15. *Reis D., Piedade B., Correia F. F., Dias J. P., Aguiar A.* Developing docker and docker-compose specifications: A developers’ survey. *IEEE Access*. 2022;10:2318–2329. EDN: JWRGVP. DOI: 10.1109/access.2021.3137671.

16. *Егоров В. Б.* Некоторые вопросы программного определения центров обработки данных. *Системы и средства информатики*. 2020;30(2):103–112. EDN: FXXGJY. DOI: 10.14357/08696527200210.

[*Egorov V. B.* Some issues of software-defined datacenters. *Systems and Means of Informatics*. 2020;30(2):103–112. (In Russian.) EDN: FXXGJY. DOI: 10.14357/08696527200210.]

17. *Roy D., Dutta M.* A systematic review and research perspective on recommender systems. *Journal of Big Data*. 2022;9:59. DOI: 10.1186/s40537-022-00592-5.

18. *He Q., Li X., Cai B.* Graph neural network recommendation algorithm based on improved dual tower model. *Scientific Reports*. 2024;14:3853. DOI: 10.1038/s41598-024-54376-3.

19. *Бруссард М.* Искусственный интеллект: пределы возможного. М.: Альпина нон-фикшн; 2020. 362 с.

[*Broussard M.* Artificial intelligence: The limits of the possible. Moscow, Alpina non-fiction; 2020. 362 p. (In Russian.)]

20. *Макшанов А. В., Журавлев А. Е., Тындыкарь Л. Н.* Большие данные. Big Data: учебник для СПО. Санкт-Петербург: Лань; 2022. 188 с.

[*Makshanov A. V., Zhuravlev A. E., Tyndykar’ L. N.* Big Data: textbook for SPE. Saint Petersburg, Lan’; 2022. 188 p. (In Russian.)]

Информация об авторах

Самылкина Надежда Николаевна, доктор пед. наук, доцент, профессор кафедры теории и методики обучения информатике, Институт математики и информатики, Московский педагогический государственный университет, г. Москва, Россия; *ORCID:* https://orcid.org/0000-0003-0797-5532; *e-mail:* nsamylkina@yandex.ru

Калинин Илья Александрович, канд. пед. наук, доцент, начальник управления информатизации, Департамент цифрового развития, Московский государственный лингвистический университет, г. Москва, Россия; доцент кафедры международной информационной безопасности, Институт информационных наук, Московский государственный лингвистический университет, г. Москва, Россия; *ORCID:* https://orcid.org/0000-0001-6710-8188; *e-mail:* kalininIlya@mail.ru

Information about the authors

Nadezhda N. Samylkina, Doctor of Sciences (Education), Docent, Professor at the Department of Theory and Methodology of Informatics Education, Institute of Mathematics and

Informatics, Moscow Pedagogical State University, Moscow, Russia; *ORCID:* https://orcid.org/0000-0003-0797-5532; *e-mail:* nsamylkina@yandex.ru

Ilya A. Kalinin, Candidate of Sciences (Education), Docent, Head of Informatization Administration, the Department of Digital Development, Moscow State Linguistic University, Moscow, Russia; Associate Professor at the Department of International Information Security, Institute of Information Sciences, Moscow State Linguistic University, Moscow, Russia; *ORCID:* https://orcid.org/0000-0001-6710-8188; *e-mail:* kalininIlya@mail.ru

Поступила в редакцию / Received: 26.06.24.
Поступила после рецензирования / Revised: 20.09.24.
Принята к печати / Accepted: 24.09.24.