УДК 04.065
DOI: 10.25559/SITITO.019.202303.633-645

Оригинальная статья

Предоставление безопасного доступа к данным наукометрических систем с использованием виртуальных частных баз данных

А. С. Козицын*, **М. А. Занчурин**
ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова», г. Москва, Российская Федерация
Адрес: 119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1
* alexanderkz@mail.ru

Аннотация

В статье рассматриваются способы обеспечения безопасного доступа к данным при интеграции наукометрических систем с внешними системами, основанные на применении виртуальных баз данных. Использование подобных механизмов разделения прав доступа к набору данных актуально для организации больших хранилищ данных, создания веб-приложений с очень большим количеством пользователей и в других областях. В отличие от механизмов разграничения доступа, реализованных на уровне логики приложения, механизмы разграничения доступа на уровне базы данных гарантируют обеспечение корректного доступа к данным даже при наличии ошибок в каком-либо из модулей системы, например ошибок написания SQL-запросов разработчиками, уязвимости переполнения буфера, SQL-инъекции и других. В работе представлен обзор основных способов реализации виртуальных баз данных с помощью меток безопасности, с использованием встроенных механизмов разграничения доступа на уровне ядра и промежуточного слоя представлений, показаны их преимущества и недостатки. Анализируется применимость указанных методов для предоставления доступа для экспорта и импорта данных в крупных информационных, в том числе наукометрических системах. Приводится описание программной реализации модуля безопасного доступа к наукометрическим данным, созданного с использованием описанных в статье технологий. Разработанная программная реализация позволяет предоставлять безопасный доступ пользователям к их наукометрическим данным для выгрузки во внешние системы анализа и построения отчетности. Результаты работы апробированы на данных наукометрической системы ИСТИНА. Представленные результаты могут использоваться при разработке масштабируемых наукометрических систем с различными уровнями доступа к информации.

Ключевые слова: виртуальные базы данных, контекст пользователя, информационные системы, наукометрические системы, безопасность, выгрузка данных, интеграция, распределенные системы

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Козицын А. С., Занчурин М. А. Предоставление безопасного доступа к данным наукометрических систем с использованием виртуальных частных баз данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19, № 3. С. 633-645. https://doi.org/10.25559/SITITO.019.202303.633-645

Original article

Mechanisms for Secure Access to Data of Scientometric Systems Using Virtual DBMS

A. S. Kozitsyn*, M. A. Zanchurin
Lomonosov Moscow State University, Moscow, Russian Federation
Address: 1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation
* alexanderkz@mail.ru

Abstract
The paper discusses ways to ensure secure access to data when integrating scientometric systems with external systems, based on the use of virtual databases. It presents descriptions of the main methods of implementing virtual databases, shows their advantages and disadvantages, and describes how to provide access for export and import of data. The concept of user context, methods of restricting his rights to access and modify data are considered. The presented results can be used in the development of scalable scientometric systems with different levels of access to information. The paper discusses several algorithms for identifying hidden links between scientific publications, journals, and conferences using statistical analysis to process bibliographic data. The paper presents the results of research on automatic finding translations of articles, ambiguity resolution when identifying authors of articles on bibliographic data, determining the degree of thematic proximity of journals and conferences. The presented results can be used for verification of data collected by scientometric systems, improving the quality of analytical processing of scientometric data, for constructing ergonomic interfaces of information systems, and for defining rules for security policies.

Keywords: virtual databases, user context, information systems, scientometric systems, security, export data, integration, distributed systems

Conflict of interests: The authors declare no conflict of interest.

For citation: Kozitsyn A.S., Zanchurin M.A. Mechanisms for Secure Access to Data of Scientometric Systems Using Virtual DBMS. *Modern Information Technologies and IT-Education.* 2023;19(3):633-645. https://doi.org/10.25559/SITITO.019.202303.633-645

1. Введение

Эффективное управление научной деятельностью в масштабах крупных научно-образовательных организаций или страны в целом возможно только с применением методов наукометрии[1]. Такие методы позволяют оценить основные тенденции в развитии науки, определить основные научные направления для финансового стимулирования и провести оценку качества осуществляемых исследований и преподавательской деятельности [1-3]. На первом этапе автоматизации процесса вычисления наукометрических показателей использовались системы ручного сбора данных на основе заполнения электронных таблиц с последующим агрегированием, например, с помощью формул в Excel. Однако такой метод не позволял достаточно эффективно осуществлять верификацию данных, обеспечивать их точность и полноту, проводить дедубликацию и автоматическую оценку качества, а также использовать методы интеллектуального анализа для автоматического построения гипотез и выявления скрытых зависимостей. Развитие систем цитирования частично позволило решить эти вопросы в области сбора информации о научных публикациях, но оценка всей остальной научной продукции по-прежнему проводилась в ручном режиме.

С целью автоматизации процесса сбора, верификации, аналитической обработки и агрегирования данных о научной деятельности в целом в организациях стали создаваться наукометрические системы [4, 5], которые позволяют автоматизировать построение наукометрических показателей [6] не только по научным публикациям, но и по участию в конференциях, деятельности диссертационных советов и редколлегий, по данным о преподавательской деятельности и руководству квалификационными работами, по участию в выполнении различных НИР и получению выдающихся научных достижений, по результатам научных отчетов и другим сведениям о деятельности научных работников и преподавательского состава организаций.

При разработке наукометрических систем требуется не только создание современных алгоритмов интеллектуального анализа и обработки данных [7, 8], методов автоматического выявления связей и взаимных зависимостей внутри совокупности информационных объектов [9, 10], в том числе с использованием онтологий [7], способов автоматизированного и автоматического сбора информации из внешних источников [11], но и реализация надежных средств защиты хранимых данных с настраиваемым разделением прав доступа между пользователями. При предоставлении прав доступа необходимо учитывать специфику исследуемой области, учитывать наличие авторских прав на хранимые в системе объекты при определении возможности их редактирования и просмотра. Выявление скрытых зависимостей между объектами при анализе больших объемов наукометрических данных позволяет решать задачи автоматического или автоматизированного построения онтологий и построения дополнительных правил при реализации политик безопасности доступа к данным. Использование подобных механизмов разделения прав доступа к набору данных актуально для организации больших хранилищ данных [12], создания веб-приложений с очень большим количеством пользователей [13] и в других областях.

Базу данных с реализованными разграничениями прав доступа на уровне ядра базы данных называют виртуальной, поскольку каждый пользователь при выполнении запросов видит свой вариант содержимого базы. Следует отметить, что термин «виртуальная база данных» в настоящее время имеет несколько значений. С распространением облачных технологий его стали использовать как синоним термина «облачная база данных» для обозначения базы данных, размещенной на виртуальной машине в облаке с возможностью миграции между различными серверами. Также этот термин используют для обозначения систем интеграции нескольких баз данных в единую систему, которая выполняет эмуляцию работы как с единой базой данных. В настоящей работе этот термин используется для обозначения возможности представления каждому пользователю своей версии данных в рамках одной СУБД. В отличие от СУБД с «забвением» (Oblivious) [14, 15], предполагается, что выполнение запросов происходит в доверенной среде.

В разделе 2 представлено сравнение методов разграничения доступа к данным в системах управления базами данных и дается описание примера их программной реализации для СУБД Oracle, в том числе технологии VPD (Virtual Private Database), выделены сильные и слабые стороны этих методов. В разделе 3 описывается разработанное авторами решение разграничения доступа к данным для наукометрической системы ИСТИНА.

2.Способы ограничения доступа к данным

Традиционные подходы к обеспечению безопасности базировались на применении ролевой модели RBAC (Role-based access control) [16], мандатной модели MAC (Mandatory Access Control) и дискреционной модели DAC (Discretionary Access Control). Преимуществом этих моделей является простота реализации и быстродействие. Но их выразительности оказывается недостаточно для реализации правил с динамическими параметрами. При реализации прав доступа в наукометрических системах необходимо учитывать наличие правил вида «статью может редактировать любой соавтор», «проект может видеть любой сотрудник из той же организации, что и участники проекта». Дополнительные правила доступа могут задаваться на основе результатов автоматического построения онтологий [17], которые могут строиться с использованием методов выявления скрытых семантических зависимостей между объектами. Реализация таких правил требует использования логического разграничения доступа ABAC (Attribute-Based Access Control)[2] [18] с удобными визуальными средства-

[1] Руководство по наукометрии: индикаторы развития науки и технологии / М. А. Акоев, В. А. Маркусова, О. В. Москалева, В. В. Писляков ; Thomson Reuters. Екатеринбург : Издательско-полиграфический центр «Издательство УРГУ», 2014. 250 с. https://doi.org/10.15826/B978-5-7996-1352-5.0000

[2] Guide to Attribute Based Access Control (ABAC) Definition and Considerations / V. C. Hu [et al.]. NIST Special Publication 800-162. National Institute of Standards and Technology, 2014. 37 p. https://doi.org/10.6028/NIST.SP.800-162

ми сопровождения [19], которые в настоящее время в значительной степени потеснили старые модели RBAC, MBAC и DAC, и является одним из наиболее перспективных и активно развивающихся направлений исследований в области обеспечения информационной безопасности.

В модели ABAC решение о предоставлении доступа зависит от значений атрибутов объектов и субъектов доступа, то есть объектов информационной системы и пользователя, запрашивающего выполнение операции. Политика безопасности представляется в виде набора правил доступа, учитывающих значения атрибутов объектов и использующих связи между ними. В рамках модели ABAC разработано множество конкретных решений [20], как узкоспециализированных, в которых вычисление значений атрибутов информационных объектов производится на языке высокого уровня и изменение политики требует изменения исходного кода приложения, так и общих, с возможностью гибкой настройки правил и возможностью автоматической проверки их непротиворечивости [19].

Механизмы разграничения доступа могут быть реализованы как на уровне логики приложения, так и на уровне базы данных. Преимуществом реализации механизмов доступа на уровне базы данных[3] является невозможность ошибочного предоставления доступа из-за ошибок в одном из модулей системы, таких как ошибки написания SQL-запросов разработчиками, уязвимости переполнения буфера, SQL-инъекции и других.

2.1. Использование меток доступа

Одним из способов разграничения прав доступа к данным на уровне строк таблиц является использование дополнительных столбцов, содержащих метки доступа. Такие столбцы должны содержать информацию об уровнях доступа, автоматически заполнятся при создании или изменении записей в таблицах и быть защищены от изменения пользователем. Подобные механизмы реализованы в различных СУБД (Oracle, PostgreSQL, Линтер и других). В СУБД Oracle такой механизм носит название OLS (Oracle Label Security)[4]. Для каждой записи в таблице и для каждого пользователя устанавливается специальная метка, в которой предусматривается возможность определения прав доступа на основании трех параметров: уровня (level); категории (compartment) и группы (group).

Уровень безопасности задается числом, и доступ пользователю предоставляется ко всем записям, которые имеют такой же уровень, как у пользователя, или ниже. Например, пользователь с уровнем доступа «Секретно» может видеть все записи с уровнями «Публичные», «ДСП», «Секретно», но не может видеть записи с уровнем «Совершенно секретно».

Категории задаются списком констант. При задании прав доступа через компонент категорий пользователь будет видеть только записи своей категории. Например, пользователь с категорией «Москва» будет видеть все записи с метками «Москва» и не будет видеть записи с метками «Новосибирск» вне зависимости от установленного уровня.

Группа позволяет задавать иерархический список констант. При этом пользователю можно предоставить право видеть только записи, находящиеся в его группе или в дочерних группах. Например, пользователь в группе «Администрация физического факультета» (АФК) будет видеть записи с метками «Лаборатория кристаллографии» (ЛК), но не будет видеть записи с метками «Кафедра теории вероятности».

Для задания меток всем записям таблицы создается дополнительный столбец, содержащий метку в формате LEVEL:COMPARTMENT1,...,COMPARTMENTn:GROUP1,...,GROUPn. При этом раздел категорий и групп может быть не заполнен. Для описанного выше примера метка может быть задана «ДСП:ЛК».

Выдача привилегий пользователям осуществляется с использованием графического интерфейса или системного пакета SA_USER_ADMIN. Метки для записей таблиц могут определяться на основе прав пользователя, который производит добавление записей в базу данных, или с использованием пакета SA_POLICY_ADMIN.APPLY_TABLE_POLICY, который позволяет описать произвольные логические правила с использованием значений атрибутов добавляемой записи, сессии пользователя, времени суток и любых других параметров.

Преимуществом такого подхода является простота описания политики безопасности. Для создания защищенной базы данных не требуется написания программного кода, и все действия могут быть выполнены через графический интерфейс. К основным недостаткам следует отнести: невозможность создания сложных логических правил для определения прав доступа к данным и необходимость создания отдельного пользователя в базе данных для каждого физического пользователя системы. Последний аспект особенно важен для систем, предоставляющих информационные услуги через порталы в Интернет.

2.2. Контекст пользователя

Контекст — это набор данных, который определяется в начале работы сессии и из соображений безопасности не может быть переопределен до конца сессии пользователя без использования дополнительных привилегий. В контекст может быть записана информация об идентификаторе пользователя, его подразделении и должности, уровне доступа, разрешенном времени работы и другие аналогичные данные. Данные контекста вычисляются отдельным и единым для всех подсистем модулем идентификации при вводе идентификационных данных пользователя до того, как пользователь получает доступ к системе. Поскольку дальнейшее изменение контекста запрещено пользователем на уровне СУБД, потенциальный злоумышленник никаким образом не может изменить контекст даже при наличии уязвимостей в отдельных модулях интерфейса системы.

Реализация работы с контекстом может базироваться на механизмах, предоставляемых ядром системы или собственными пакетами.

В ядре СУБД Oracle существует предопределенный контекст userenv, содержащий атрибуты с типом аутентификации кли-

[3] Баранчиков А. И., Баранчиков П. А., Пылькин А. Н. Алгоритмы и модели ограничения доступа к записям БД : монография. М. : Горячая линия-Телеком, 2016. 182 с. URL: https://e.lanbook.com/book/111010 (дата обращения: 27.08.2023).

[4] Oracle Label Security [Электронный ресурс] // Oracle, 2023. URL: https://docs.oracle.com/en/database/oracle/oracle-database/19/olsag/index.html (дата обращения: 27.08.2023).

ента, ip-адреса клиента, именем пользователя в операционной системе, схемы данных и другие параметры сессии, которые можно использовать для определения правил безопасности. Значение атрибутов определяется ядром системы и не может быть изменено пользователем. Также можно определить свой контекст и указать программный модуль, который может его изменять. Например, CREATE CONTEXT myCon USING myProc. При таком определении команда задания переменных контекста DBMS_SESSION.SET_CONTEXT ('myCon', 'Param1', 'value1') может быть вызвана только из процедуры myProc, а считана в любом SQL-выражении вызовом функции sys_context ('myCon', 'Param1'). При использовании собственных пакетов в теле пакета определяется набор переменных контекста, которые могут быть перезаписаны только процедурами этого пакета. Из соображений безопасности возможность вызова пользователем процедур перезаписи контекста или переменных пакета должна быть ограничена.

Существует два подхода для реализации защиты содержимого контекста от изменения пользователем в процессе работы: использование специального пользователя или пакета для определения контекста и использование пакетов с возможностью установки контекста, но без возможности его перезаписи. В первом случае команды определения контекста могут вызываться только со специальными правами. Одним из распространенных способов такой защиты является использование триггера класса after logon on database, который вызывается под правами суперпользователя и в котором может происходить вызов процедуры вычисления всех необходимых параметров и установки контекста.

Преимуществом такого подхода является простота и гарантированность защиты. Триггер такого класса выполняется в момент соединения до начала обработки поступающих от пользователя команд. Таким образом, потенциальный злоумышленник не может повлиять на его работу, даже получив прямой доступ к базе с возможностью выполнения любых команд DML и DDL. Недостатком является невозможность передачи в контекст запрашиваемых у пользователя параметров. Это особенно важно для веб-приложений, которые имеют очень большое количество пользователей, имеют собственные средства аутентификации и предоставляют всем сессиям доступ к базе данных под единой учетной записью СУБД. В таких приложениях на основе данных о прохождение аутентификации для определения контекста требуется передать данные аутентификации в базу данных уже после установки соединения. В этом случае использование механизма триггеров невозможно, и требуется создавать процедуры с иными способами ограничения доступа.

Эффективным способом ограничения является невозможность перезаписи установленного контекста. При получении команды изменения переменной контекста процедура проверяет ее текущее значение и, если такое значение уже было установлено ранее, не производит новую запись. При реализации такого варианта защиты очень важно следить, чтобы приложения не могли предоставить доступ к системе пользователю, у которого еще не установлены все переменные контекста. Для этого создается единый для всех приложений сервис авторизации, код которого тщательно проверяется на наличие потенциальных уязвимостей. Перед началом работы любого приложения пользователю предлагается задать свои учетные данные через этот сервис, сервис на основе полученных данных производит аутентификацию клиента и устанавливает все переменные контекста и только после этого передает управление приложению. Таким образом, даже при эксплуатации злоумышленником уязвимостей приложения переопределить контекст сессии будет невозможно.

2.3. Ограничение на уровне ядра

Использование такого подхода требует определения атрибутов или связей между объектами. Часть отношений может задаваться явно при вводе данных в систему пользователями. Например, при регистрации в наукометрической системе статьи пользователи обычно указывают авторов статьи и журнал. Однако значительная часть отношений не вводится в систему пользователем. Например, зависимости между публикациями, зависимости между журналами, связи между авторами, которые существуют в реальном мире, но не описываются пользователями в явном виде при вводе данных. Выявление скрытых отношений и их использование при описании политик безопасности позволяет расширить возможности для описания правил доступа к данным.

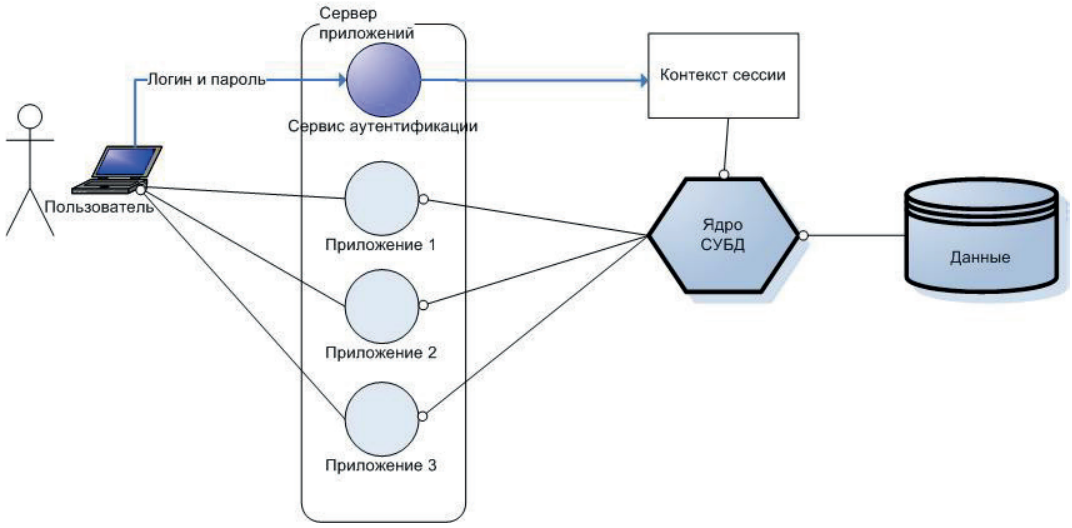
Для задания ограничений на основе выделенных связей могут использоваться механизмы, реализованные на уровне ядра СУБД [21]. Такие механизмы не требуют внесения изменений в существующий программный код унаследованных приложений. В СУБД PostgreSQL для реализации подобного вида ограничений разработан механизм Row Security Policies [5], в СУБД MSQL реализован механизм Row-Level Security [6].

Для иллюстрации возможностей такого подхода рассмотрим реализацию механизма Virtual Private Database(VPD) фирмы Oracle (Рис. 1), которая первой представила подобный функционал в промышленной версии своего продукта[7].

[5] Описание механизма RSL в PostgreSQL [Электронный ресурс] // The PostgreSQL Global Development Group, 2023. URL: https://www.postgresql.org/docs/9.5/ddl-rowsecurity.html (дата обращения: 27.08.2023).

[6] Описание механизма RSL в MSSQL [Электронный ресурс] // Quest Software Inc. A, 2023. URL: https://www.sqlshack.com/introduction-to-row-level-security-in-sql-server/ (дата обращения: 27.08.2023).

[7] Описание механизма VPD в Oracle [Электронный ресурс] // Oracle, 2023. URL: https://docs.oracle.com/cd/B28359_01/network.111/b28531/vpd.htm#DBSEG007 (дата обращения: 27.08.2023).



Р и с. 1. Схема работы с VPD
F i g. 1. Scheme of work with VPD

Источник: здесь и далее в статье все таблицы и рисунки составлены авторами.
Source: Hereinafter in this article all tables and figures were made by the authors.

Основными объектами, с которыми работает механизм VPD, являются: защищаемые таблицы; контекст пользователя и функции определения предикатов. Задачей VPD является надежное ограничение видимости строк в защищаемых таблицах в зависимости от контекста пользователя, например, от его места работы, должности или других определяемых сервисом авторизации признаков. При осуществлении запроса к каждой таблице базы данных ядро системы вызывает функцию определения дополнительного предиката безопасности для этой таблицы, и полученное выражение добавляется к запросу. Например, если пользователю разрешено просматривать статьи только за 2021 год, то к каждому запросу с использованием таблицы Article будет добавляться условие f_article_year=2021, и пользователь никаким способом не сможет получить из таблицы записи о других статьях. Функционал VPD в СУБД Oracle позволяет гибко определять предикат для таблицы в зависимости от контекста пользователя. Также различные виды предикатов могут определяться для операций выборки данных (select), изменения данных (insert/update/delete) или индексации (index). Добавление политики безопасности осуществляется функцией DBMS_RLS.ADD_POLICY, которая позволяет определять функцию вычисления предиката для каждой таблицы. Предикат политики может действовать не на всю защищаемую таблицу, а только на определенные столбцы, перечисленные в параметре sec_relevant_cols. В этом случае при выполнении запроса на чтение пользователь увидит строки таблицы с пустыми значениями в скрытых от него столбцах. Условие защиты столбцов может также зависеть от предиката таблицы. Например, можно скрыть суммы финансирования проектов, если проект финансируется не из РНФ. Кроме того, ядро позволяет проводить автоматическую проверку, что при изменении значений строки она не окажется скрытой от пользователя (параметр создания политики update_check).

Основным недостатком такого механизма является увеличение нагрузки на ядро СУБД и уменьшение производительности системы и требует применения более сложных методов оптимизации производительности[8]. Частично проблема с производительностью решается наличием возможности выбирать способ кэширования предиката (policy_type). Существуют пять видов функций предикатов: статический (STATIC); статический разделяемый (SHARED_STATIC); контекстный (CONTEXT_SENSITIVE); контекстный разделяемый (SHARED_CONTEXT_SENSITIVE) и динамический (DYNAMIC).

Текст предиката статических функций вычисляется только один раз и хранится в кэше СУБД. Это позволяет ускорить вычисление запроса, но может привести к нарушению политики безопасности, если текст предиката должен зависеть, например, от логина пользователя.

Динамические функции для определения контекста вызываются при каждом обращении пользователя к таблице. Текст предиката при таком вызове учитывает все параметры пользователя и среды, например, времени суток. Однако в большинстве случаев это приводит к излишним вычислениям, поскольку, как правило, значение предиката определяется только контекстом пользователя.

Значения предикатов для контекстных функций при кэшировании учитывают контекст и пересчитываются при изменении контекста. По умолчанию сравнение производится по всем атрибутам контекстов. Однако при создании политики безопасности можно указать в параметрах namespace и attribute конкретный контекст и атрибут, который будет определять политику кэширования значений функции вычисления предикатов.

[8] Кайт Т., Дарл К. Oracle для профессионалов. Технологии и решения для достижения высокой производительности и эффективности. М.: Вильямс, 2015. 960 с.

Также следует отметить, что использование политик безопасности виртуальных частных баз данных не позволяет использовать прямые механизмы доступа к данным (direct-path operations): прямую загрузку в SQL*Loader; прямую вставку (Direct Path Insert) с использованием подсказки APPEND (INSERT /*+ APPEND */ INTO ...) или прямой экспорт. Подобные ограничения вызваны тем обстоятельством, что механизм предикатов работает на уровне SQL, а прямой доступ к данным производится на более низком уровне. Поэтому при попытке его использования для защищаемых таблиц ядро СУБД вернет ошибку, если активному пользователю СУБД не предоставлена привилегия EXEMPT ACCESS POLICY.

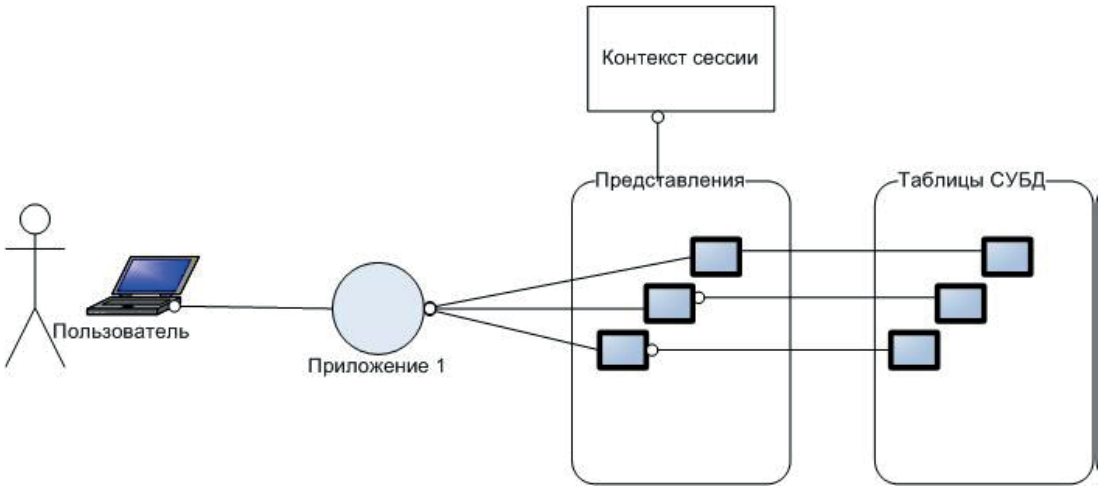
Следует отметить, что, несмотря на прозрачность такого способа защиты данных с точки зрения разработчиков приложений, такой подход имеет существенный недостаток. Способ программной реализации и настройки этого механизма в различных СУБД (Oracle, PostreSQL, MSSQL) существенно отличаются, что значительно осложняет процесс миграции разрабатываемой программной системы между этими СУБД.

2.4. Ограничение на уровне представлений

Реализация механизма разграничения доступа на уровне контекста может производиться с использованием создания представлений. Несмотря на то, что этот механизм[9] использовался во многих базах данных до появления в них встроенной поддержки ограничений на уровне ядра, он продолжает оставаться востребованным и на сегодняшний день.

Для реализации такой схемы в СУБД создается два пользователя. Первый пользователь (владелец данных) назначается владельцем всех таблиц данных, а также в нем создаются представления к каждой таблице данных, в предикатах которых указаны права доступа к данным с учетом переменных контекста.

Второму пользователю (будем в дальнейшем называть его пользователем приложений) выдаются права на обращения только к представлениям (Рис. 2). Все приложения подсоединяются к базе данных только с правами второго пользователя. Таким образом, даже при установке полного контроля над кодом приложения злоумышленник не сможет прочитать из СУБД данные, которые ему не разрешены в соответствии с его контекстом.



Р и с. 2. Схема работы с промежуточным слоем представлений

F i g. 2. Scheme for work with the intermediate presentation layer

Основным преимуществом такого подхода является возможность построения дополнительных представлений для часто используемых соединений с целью увеличения производительности СУБД.

Еще одним преимуществом данного подхода является независимость от проприетарных технологий. Его программная реализации практически не зависит от вида СУБД, что значительно облегчает процесс миграции приложений между различными видами СУБД, например с Oracle на PostreSQL.

Основным недостатком такой схемы является необходимость дублирования всех таблиц в базе данных в виде представлений, поддержка соответствия структуры таблиц и представлений при модификации структуры данных и создание специальных процедур или триггеров для реализации модификации самих данных, если необходимо обеспечивать доступ не только на изменение, но и на модификацию данных. Кроме того, в некоторых случаях, например в унаследованных системах, бывает трудно осуществить перевод всех приложений на отдельного пользователя приложений.

9 Rask A., Rubin D., Neumann B. Implementing Row- and Cell-Level Security in Classified Databases Using SQL Server 2005 [Электронный ресурс] // Microsoft Learn, 2023. URL: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2005/administrator/cc966395(v=technet.10)?redirectedfrom=MSDN (дата обращения: 27.08.2023).

2.5. Сравнение методов

Преимущества и недостатки рассмотренных выше методов обеспечения защиты данных на уровне базы данных приведены в следующих Таблицах 1-3.

Таблица 1. **Сравнение методов хранения контекстов**
Table 1. **Comparison of context storage methods**

	Переменные пакета	sys_context
Проприетарность	-	+
Уровень ядра	-	+
Возможность вычисляемых значений	+	-
Необходимость дополнительного написания кода	+	-

Таблица 2. **Сравнение методов задания контекстов**
Table 2. **Comparison of methods for setting contexts**

	В пакете	В триггере
Проприетарность	-	+
Уровень ядра	-	+
Возможность использования дополнительных данных сессии	+	-
Необходимость дополнительного написания кода	+	-

Таблица 3. **Сравнение методов защиты данных**
Table 3. **Comparison of data protection methods**

	Слой представлений	Virtual Private Database
Проприетарность	-	+
Уровень ядра	-	+
Виды предикатов	Статические	Статически, динамические, контекстные
Необходимость дополнительного написания кода	+	-
Возможность оптимизации предикатов для часто выполняемых соединений	+	-

При построении систем доступа к наукометрическим данным и созданию API для выгрузки наиболее значимыми критериями являются:
- возможность оптимизации предикатов для часто выполняемых соединений, поскольку данные в наукометрических системах имеют сложную структуру взаимных зависимостей;
- возможность использования дополнительных данных сессии, которые определяются сервисом идентификации;
- возможность перехода на непроприетарные СУБД (например, Postgresql) для выполнения требований, предъявляемым к критически важным системам в РФ.

Исходя из этих критериев и представленных в Таблицах 1-3 данных, для разграничения прав доступа к данным в наукометрических системах наиболее оптимальным является реализация механизмов поддержки контекста в переменных пакета с обеспечением процесса инициализации на уровне этого пакета и использование для защиты данных промежуточного слоя представлений.

Оптимизация предикатов обеспечивается в сложных запросах созданием агрегирующих представлений для описаний для выгрузки описания объектов. Например, если при выгрузке данных из наукометрической системы требуется часто выбирать описания статей авторов, работающих в определенном подразделении, а политика безопасности позволяет пользователю выгружать только статьи сотрудников своего подразделения. При использовании встроенных в ядро базы данных механизмов разграничения доступа необходимо было бы выполнить три предиката: для статей, для авторов и для мест работы. При создании отдельного представления для выгрузки данных достаточно будет выполнить проверку только по одному наименее ресурсоемкому предикату — предикату мест работы. Все остальные предикаты в этом случае оказываются избыточными из-за последующего соединения таблиц в тексте основного запроса. Таким образом, при выгрузке большого количества данных по связанным таблицам и использовании дополнительных представлений удается получить ускорение запросов в десятки раз.

Возможность использования дополнительных данных сессии с использованием переменных пакетов обеспечивается процессом авторизации, который проверяет идентификационные данные пользователя и по ним определяет необходимые дополнительные свойства пользователя, например место работы.

Использование при создании представлений синтаксиса, который поддерживается Oracle и PostreSQL, позволяет легко переносить текст представлений между этими СУБД, что может являться критическим фактором при внедрении системы в крупных государственных организациях.

3. Предоставление доступа к наукометрическим данным

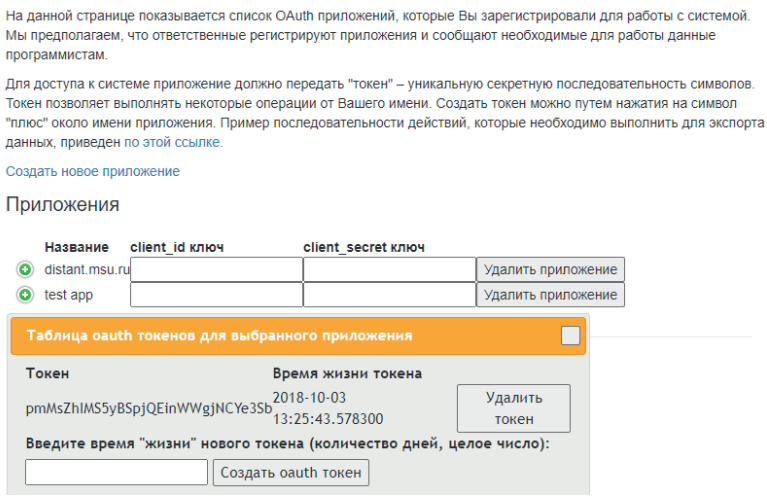
Проведение анализа основных трендов в научной сфере и построение аналитических отчетов на основе данных наукометрических систем [22, 23] требует создания механизмов безопасной выгрузки значительной части доступных пользователю наукометрических данных. Для организации безопасного доступа к данным в информационной системе в первую очередь должен быть организованы контекст с возможностью их разовой установки в начале сессии и последующем доступе в режиме только чтения в течение всего времени работы пользователя. Для наукометрических систем контекст должен включать в себя информацию о пользователе и организации. Кроме того, для ускорения проверки прав доступа в процессе работы могут дополнительно задаваться подразделение, должность и другие параметра. В программной реализации этого метода, которая была разработана авторами для апробации на данных системы ИСТИНА, для работы контекстом используется пакет SEC_ORG.

Первичная идентификация пользователей осуществляется

реализованным на уровне приложения механизмом авторизации OAuth. OAuth является открытым протоколом авторизации пользовательских приложений, который позволяет предоставить третьей стороне ограниченный доступ к защищенным ресурсам пользователя без необходимости передачи третьей стороне логина и пароля пользователя. При выборе механизмов авторизации был выбран именно OAuth, так как была поставлена задача предоставить возможность предоставлять доступ к внешнем API наукометрической системы без использования логина/пароля пользователя при каждом начале новой сессии.

Первичная регистрация пользователя в системе происходит стандартным методом. Пользователь проходит аутентификацию с помощью ввода логина и пароля. Далее, используя раздел для пользователей с повышенными правами, пользователь осуществляет создание, регистрацию и управление токенами и приложениями OAuth (Рис. 3). Пользователи могут создавать свои OAuth-приложения в интерфейсе системы и далее генерировать токены с указанным временем жизни. Каждый созданный токен может использоваться для доступа к ресурсам системы и, в частности, для доступа к API методам для выгрузки данных. Перед началом сессии пользователь передает значение токена приложению наукометрической системы. Получив токен, приложение системы определяет, какому OAuth-приложению принадлежит переданный токен. Далее происходит определение, какой пользователь создал данное приложение, и информация о пользователе используется для создания контекста при начале сессии.

Авторами статьи внедрены механизмы аутентификации и реализован пользовательский интерфейс для удобного создания токенов доступа OAuth в функционале наукометрической системы организации.



Р и с. 3. Интерфейс создания OAuth-токенов

F i g. 3. Interface for creating OAuth tokens

Разработанный для этих целей интерфейс (Рис. 3) позволяет создавать токены OAuth и указывать им время жизни. На рисунке представлен пример страницы наукометрической системы с созданным токеном, у которого уже закончилось время жизни.

Для использования аутентификации с помощью OAuth-токена необходимо добавить заголовок в HTTP запрос вида Authorization: Bearer access_token. Далее представлены примеры запросов к API-методам:

- **Создание задачи на выгрузку данных**

Запрос: curl -H “Authorization: Bearer AykGU2IbJ12g2Q7zCEPIQxg0q8s60x” \
-X GET https://somedomain/api/applications/export/?object_type=ARTICLE
Ответ: {“result”: {“task_id”: “6ed005e3-7f8e-4863-a457-184e6e5aea27”}}

- **Проверка статуса выполнения задачи**

Запрос: curl -H “Authorization: Bearer AykGU2IbJ12g2Q7zCEPIQxg0q8s60x” \
-X GET “https://somedomain/api/tasks/state/?task_id=6ed005e3-7f8e-4863-a457-184e6e5aea27”
Ответ: {“result”: {“status”: “SUCCESS”, “task_id”: “6ed005e3-7f8e-4863-a457-184e6e5aea27”}}

- **Получение результатов выполнения задачи**

Запрос: curl -H “Authorization: Bearer AykGU2IbJ12g2Q7zCEPIQxg0q8s60x” \
-X GET “https://somedomain/api/tasks/result/?task_id=6ed005e3-7f8e-4863-a457-184e6e5aea27”

Представленные методы реализуют асинхронный механизм доступа к данным наукометрической системы. При создании задачи выгрузки данных она ставится в очередь, которая обрабатывается по мере освобождения ресурсов системы. Периодически внешняя система проверяет состояние поставленной задачи и после получения сообщения о готовности отправляет запрос на получение результатов. Такой подход позволяет избежать перегрузки системы в «пиковые» моменты спроса на ресурсы и гарантировать выполнение каждого запроса.

Доступ к данным осуществляется через промежуточный слой

представлений с использованием установленного на этапе соединения контекста. Для улучшения читаемости структуры БД рекомендуется создавать представления с использованием префиксов или суффиксов, которые добавляются к имени основной таблицы БД информационной системы. Например, если основная таблица называется ARTICLE, то представление для этой таблицы на основе контекста называется ARTICLE_ORG. Для предоставления данных авторизованному пользователю должны предоставляться два метода: метод получения списка доступных данных и метод выгрузки данных. Для апробации на наукометрической системе ИСТИНА использовались следующие методы.

Метод предоставления списка доступных для получения данных из наукометрической системы IMP_ORG.get_enable_list. Данный метод предоставляет пользователю или стороннему приложению список данных, к которым можно получить доступ на чтение с применением внешнего API. В настоящей реализации этот список состоит из следующих пунктов:

- ARTICLE — статьи сотрудников организации;
- ARTIRID — статьи сотрудников организации с указанием уникального идентификатора системы для каждого автора (IRID) [24, 25];
- CHAPTER — главы в монографиях, опубликованных сотрудниками организации;
- BOOK — опубликованные сотрудниками организации книги;
- BKIRID — опубликованные сотрудниками организации книги с IRID;
- MANS — список сотрудников организации;
- AWARD — список полученных сотрудниками организации наград;
- PRESENTATION — сделанные сотрудниками организации доклады;
- BOARD — список диссертационных советов организации;
- DISSER — список защищенных сотрудниками организации диссертаций;
- PATENT — список полученных сотрудниками организации патентов;
- PATPROGRAM — список полученных сотрудниками организации свидетельств о регистрации ПО;
- PROJECT — информация по проектам организации;
- EDUCWORK — перечень педагогической нагрузки, выполняемой сотрудниками организации.

Список доступных методов может модифицироваться и расширяться в зависимости от особенностей конкретной наукометрической системы, в которой используется этот подход предоставления данных для интеграции с другими системами. После получения списка доступных данных пользователь или внешнее приложение могут использовать метод IMP_ORG.get_import, позволяющий получить данные в формате XML. Метод имеет следующие параметры:

- list — список запрашиваемых данных (например, ‘ARTICLE,PATENT’);
- mans_id — идентификатор сотрудника IRID, по которому требуется получить данные (может не совпадать с IRID владельца токена);
- department_id — ключ подразделения организации;
- begin_year — начало выгружаемого периода;
- end_year — конец выгружаемого периода;
- mans_list — позволяет задать список сотрудников IRID, по которым требуется выгрузить данные.

Параметры mans_id, department_id, begin_year, end_year, mans_list являются необязательными и служат для обеспечения возможности дополнительной фильтрации выгружаемых данных. Например, если требуется выгрузить все статьи за период с 2018 по 2020 год, опубликованные сотрудниками механико-математического факультета, то в метод IMP_ORG.get_import передаются следующие параметры: (list=>ARTICLE, begin_year=>2018, end_year=>2020, department_id=>275889), где 275889 — ключ механико-математического факультета МГУ имени М.В. Ломоносова.

Полученные данные в виде XML возвращаются пользователю для загрузки в собственную аналитическую систему, выгрузки на сайт, использования в других информационных системах или создания архива. На рисунке 4 представлен пример предоставляемых пользователю данных. Формат и список полей выгружаемого XML зависит от вида запрашиваемых данных.

<?xml version="1.0" encoding="UTF-8"?>
<response><pam:message xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:pam="http://prismstandard.org/namespaces/pam/2.0/" xmlns:prism="http://prismstandard.org/namespaces/basic/2.0/" xmlns:xhtml="http://www.w3.org/1999/xhtml"><xhtml:head>
<pam:article><dc:title>The information-expert system for investigation of structural and mechanical properties of materials</dc:title><dc:creator>Vasenin V.A.</dc:creator><dc:creator>Vasin R.A.</dc:creator><dc:creator>Kozitsyn A.S.</dc:creator><dc:creator>Shundeev A.S.</
<dc:publisher>Pleiades Publishing, Ltd</dc:publisher><prism:publicationDate>2005</prism:publicationDate>
<prism:volume>66</prism:volume><prism:number>7</prism:number><prism:startingPage>1177</prism:startingPage><prism:endingPage>1184</prism:endingPage><prism:url>http://istina.msu.ru/publications/article/2862482/</prism:url></pam:article></xhtml:head>
<xhtml:body><h1>Abstract</h1><p>This paper is devoted to the presentation of approaches applied to the development of an information-expert system for the investigation of structural-mechanical properties of materials. The demand for such a system follows from the necessity of th
</xhtml:body>
</pam:message><pam:message xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:pam="http://prismstandard.org/namespaces/pam/2.0/" xmlns:prism="http://prismstandard.org/namespaces/basic/2.0/" xmlns:xhtml="http://www.w3.org/1999/xhtml"><xhtml:head>
<pam:article><dc:title>Индексация коллекций документов в поисковой системе АСТАИ</dc:title><dc:creator></dc:creator><dc:creator></dc:creator><dc:creator></dc:creator><dc:creator></dc:creator><prism:publicationName>Прикладная математика и механика</prism:publicationName>
<dc:publisher></dc:publisher><prism:publicationDate>2008</prism:publicationDate>
<prism:volume>11</prism:volume><prism:url>http://istina.msu.ru/publications/article/211525/</prism:url></pam:article></xhtml:head>
<xhtml:body>
</xhtml:body>
</pam:message><pam:message xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:pam="http://prismstandard.org/namespaces/pam/2.0/" xmlns:prism="http://prismstandard.org/namespaces/basic/2.0/" xmlns:xhtml="http://www.w3.org/1999/xhtml"><xhtml:head>
<pam:article><dc:title> Информационно-экспертная система для исследования структурных и механических свойств материалов</dc:title><dc:creator></dc:creator><dc:creator></dc:creator><dc:creator></dc:creator><dc:creator></dc:creator><prism:publicationName>Ав
<dc:publisher></dc:publisher><prism:publicationDate>2005</prism:publicationDate><prism:number>7</prism:number><prism:startingPage>171</prism:startingPage><prism:endingPage>179</prism:endingPage><prism:url>http://istina.msu.ru/publications/article/211481/</
<xhtml:body>
</xhtml:body>
</pam:message><pam:message xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:pam="http://prismstandard.org/namespaces/pam/2.0/" xmlns:prism="http://prismstandard.org/namespaces/basic/2.0/" xmlns:xhtml="http://www.w3.org/1999/xhtml"><xhtml:head>
<pam:article><dc:title>Определение тематической близости научных журналов и конференций с использованием анализа графа соавторства</dc:title><dc:creator>Козицын А.С.</dc:creator><dc:creator>Афонин С.А.</dc:creator><dc:creator>Шачнев Д.А.</dc:creator><
<dc:publisher>Федеральное государственное автономное образовательное учреждение высшего образования "Казанский (Приволжский) федеральный университет"</dc:publisher><prism:publicationDate>2020</prism:publicationDate>
<prism:volume>23</prism:volume><prism:number>3</prism:number><prism:startingPage>514</prism:startingPage><prism:endingPage>525</prism:endingPage><prism:url>http://istina.msu.ru/publications/article/315089185/</prism:url></pam:article></xhtml:head>
<xhtml:body>
</xhtml:body>
</pam:message><pam:message xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:pam="http://prismstandard.org/namespaces/pam/2.0/" xmlns:prism="http://prismstandard.org/namespaces/basic/2.0/" xmlns:xhtml="http://www.w3.org/1999/xhtml"><xhtml:head>
<pam:article><dc:title>АЛГОРИТМ ОПРЕДЕЛЕНИЯ ПЕРЕВОДОВ СТАТЕЙ С ИСПОЛЬЗОВАНИЕМ СТАТИСТИЧЕСКИХ ДАННЫХ</dc:title><dc:creator>КОЗИЦЫН Александр Сергеевич</dc:creator><dc:creator>АФОНИН Сергей Александрович</dc:creator><dc:creator>ЗЕНЗИНОВ А
<dc:publisher>Федеральное государственное автономное образовательное учреждение высшего образования "Казанский (Приволжский) федеральный университет"</dc:publisher><prism:publicationDate>2018</prism:publicationDate>
<prism:volume>21</prism:volume><prism:number>6</prism:number><prism:startingPage>494</prism:startingPage><prism:endingPage>505</prism:endingPage><prism:url>http://istina.msu.ru/publications/article/170550432/</prism:url></pam:article></xhtml:head>
<xhtml:body>
</xhtml:body>

Р и с. 4. Пример XML выгружаемого списка статей

F i g. 4. Example XML of a discardable list of articles

Выбор такой архитектуры API выгрузки данных позволяет по мере необходимости добавлять как новые типы выгружаемых данных, так и новые поля в существующие выгрузки без нарушения работы унаследованных приложений. Возможность фильтрации по типам данных, по диапазонам, сотрудникам и подразделениям позволяет сторонним приложениям получать только те данные ,которые им требуются для работы, не создавая лишней нагрузки на сетевые и вычислительные ресурсы.

4. Заключение

Описанные в настоящей работе методы создания и использования виртуальной базы данных позволяет осуществлять эффективную защиту данных на уровне СУБД, что исключает возможность несанкционированного доступа к данным через эксплуатацию уязвимостей на уровне приложений. Программная реализация метода были апробирована при разработке наукометрической системы МГУ имени М.В. Ломоносова [4] и эффективно используется для интеграции ресурсов центральной наукометрической системы с локальными системами отдельных факультетов.

Список использованных источников

[1] Национальная научно-информационная инфраструктура: проблемы, задачи и перспективы / А. Е. Гуськов, А. С. Карауш, И. Е. Меньщиков [и др.] // Управление наукой и наукометрия. 2022. Т. 17, № 3. С. 380-407. https://doi.org/10.33873/2686-6706.2022.17-3.380-407

[2] Орлов А. И. Наукометрия и управление научной деятельностью // Управление большими системами. Специальный выпуск 44: Наукометрия и экспертиза в управлении наукой. 2013. № 44. С. 538-568. EDN: RDQBUH

[3] Бричковский В. В. Наукометрический анализ в информационном обеспечении инновационной деятельности // Наука и инновации. 2017. № 8(174). С. 64-67. EDN: ZIWGKN

[4] Садовничий В. А., Васенин В. А. Интеллектуальная система тематического исследования наукометрических данных: предпосылки создания и методология разработки. Часть 1 // Программная инженерия. 2018. Т. 9, № 2. С. 51-58. https://doi.org/10.17587/prin.9.51-58

[5] Архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования / В. А. Васенин, М. А. Занчурин, А. С. Козицын [и др.] // Программная инженерия. 2017. Т. 8, № 10. С. 448-455. https://doi.org/10.17587/prin.8.448-455

[6] Васенин В. А., Зензинов А. А., Лунев К. В. Использование наукометрических информационно-аналитических систем для автоматизации проведения конкурсных процедур на примере информационно-аналитической системы ИСТИНА // Программная инженерия. 2005. Т. 7, № 10. С. 472-480. https://doi.org/10.17587/prin.7.472-480

[7] Афонин С. А., Козицын А. С., Шачнев Д. А. Программные механизмы агрегации данных, основанные на онтологическом представлении структуры реляционной базы наукометрических данных // Программная инженерия. 2016.Т. 7, № 9. С. 408-413. https://doi.org/10.17587/prin.7.408-413

[8] Козицын А. С., Афонин С. А., Шачнев Д. А. Метод оценки тематической близости научных журналов // Программная инженерия. 2020. № 6. С. 335-341. https://doi.org/10.17587/prin.11.335-341

[9] Козицын А. С., Афонин С. А., Зензинов А. А. Алгоритм определения переводов статей с использованием статистических данных // Электронные библиотеки. 2018. Т. 21, № 6. С. 494-505. EDN: VVFJBM

[10] Козицын А. С., Афонин С. А. Алгоритм разрешения неоднозначности имен авторов в ИАС ИСТИНА // Современные информационные технологии и ИТ-образование. 2020. Т. 16, № 1. С. 108-117. https://doi.org/10.25559/SITITO.16.202001.108-117

[11] Методы и программные средства для ввода и верификации наукометрических данных / А. А. Зензинов [и др.] // CEUR Workshop Proceedings. 2019. Т. 2514. С. 356-367. URL: https://ceur-ws.org/Vol-2514/paper67.pdf (дата обращения: 27.08.2023).

[12] Security Concerns in Data Warehouses: Implementation and Analysis of Virtual Private Database / V. Tripathi [et al.] // 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom). New Delhi, India: IEEE Computer Society, 2019. P. 117-120. URL: https://ieeexplore.ieee.org/document/8991323 (дата обращения: 27.08.2023).

[13] Chen H.-Z. Oracle HTML DB Application with Virtual Private Database // System Simulation Technology. 2006. Vol. 2, no. 4. P. 244-248.

[14] Eskandarian S., Zaharia M. ObliDB: Oblivious Query Processing for Secure Databases // Proceedings of the VLDB Endowment. 2019. № 13(2). P. 169-183. https://doi.org/10.14778/3364324.3364331

[15] Han Z., Hu H. ProDB: A memory-secure database using hardware enclave and practical oblivious RAM // Information Systems. 2021. Vol. 96. Article number: 101681. https://doi.org/10.1016/j.is.2020.101681

[16] Role-Based Access Control Models / R. S. Sandhu [et al.] // Computer. 1996. Vol. 29, no. 2. P. 38-47. https://doi.org/10.1109/2.485845

[17] Afonin S. Ontology Models for Access Control Systems // 2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC). Vladivostok, Russia: IEEE Computer Society, 2018. P. 1-6. https://doi.org/10.1109/RPC.2018.8482178

[18] Jin X., Krishnan R., Sandhu R. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC // Data and Applications Security and Privacy XXVI. DBSec 2012. Lecture Notes in Computer Science ; ed. by N. Cuppens-Boulahia, F. Cuppens, J. Garcia-Alfaro. Vol. 7371. Berlin, Heidelberg: Springer, 2012. P. 41-55. https://doi.org/10.1007/978-3-642-31540-4_4

[19] Васенин В. А., Явтушенко Е. Д. Средства сопровождения процессов разграничения доступа к большим наукометрическим данным с использованием механизмов визуального представления // Программная инженерия. 2020. № 3. С. 131-141. https://doi.org/10.17587/prin.11.131-141
[20] Servos D., Osborn S. L. Current Research and Open Problems in Attribute-Based Access Control // ACM Computer Surveys. 2017. Vol. 49, no. 4. Article number: 65. https://doi.org/10.1145/3007204
[21] Spendolini S. Virtual Private Database // Expert Oracle Application Express Security. Apress, Berkeley, CA, 2013. P. 211-223. https://doi.org/10.1007/978-1-4302-4732-6_12
[22] Liu J., Li X., Wang Sh. What have we learnt from 10 years of fintech research? A scientometric analysis // Technological Forecasting and Social Change. 2020. Vol. 155. Article number: 120022. https://doi.org/10.1016/j.techfore.2020.120022
[23] Visualizing the emerging trends of biochar research and applications in 2019: a scientometric analysis and review / P. Wu [et al.] // Biochar. 2020. Vol. 2. P. 135-150. https://doi.org/10.1007/s42773-020-00055-1
[24] Афонин С. А., Гаспарянц А. Э. Автоматическое построение функции оценки качества в задаче разрешения неоднозначности имен авторов научных публикаций // Программная инженерия. 2015. № 10. С. 31-37. EDN: UNEXBN
[25] Козицын А. С., Афонин С. А. Разрешение неоднозначностей при определении авторов публикации с использование графов соавторства в больших коллекциях библиографических данных // Программная инженерия. 2017. Т. 8, № 12. С. 556-562. https://doi.org/10.17587/prin.8.556-562

Поступила 27.08.2023; одобрена после рецензирования 30.09.2023; принята к публикации 09.10.2023.

Об авторах:
Козицын Александр Сергеевич, ведущий научный сотрудник Научно-исследовательского института механики, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), кандидат физико-математических наук, **ORCID: https://orcid.org/0000-0002-8065-9061**, alexanderkz@mail.ru
Занчурин Максим Анатольевич, младший научный сотрудник, Научно-исследовательского института механики, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), **ORCID: https://orcid.org/0009-0006-2195-0920**, maxim.zanchurin@gmail.com

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

[1] Guskov A.E., Karaush A.S., Menshchikov I.E., Shkolin A.V., Nedelskiy V.O., Sabirov D.S., Shchukin T.N. National Scientific Information Infrastructure: Problems, Tasks and Prospects. *Science Governance and Scientometrics.* 2022;17(3):380-407. (In Russ., abstract in Eng.) https://doi.org/10.33873/2686-6706.2022.17-3.380-407
[2] Orlov A.I. *Naukometriia i upravlenie nauchnoi deiatelnostiu* [Scientometrics and Research Management]. *Large-Scale Systems Control.* 2013;(44):538-568. (In Russ., abstract in Eng.) EDN: RDQBUH
[3] Brichkovskii V.V. *Naukometricheskii analiz v informatsionnom obespechenii innovatsionnoi deiatelnosti* [Scientometric analysis in the information support of innovation activities]. *The Science and Innovations.* 2017;(8):64-67. (In Russ., abstract in Eng.) EDN: ZIWGKN
[4] Sadovnichii V.A., Vasenin V.A. *Intellektualnaia sistema tematicheskogo issledovaniia naukometricheskikh dannykh: predposylki sozdaniia i metodologiia razrabotki. Chast' 1* [An intelligent system for the case study of scientometric data: prerequisites for creation and development methodology. Part 1]. *Programmnaia inzheneriia* = Software Engineering. 2018;9(2):51-58. (In Russ., abstract in Eng.) https://doi.org/10.17587/prin.9.51-58
[5] Vasenin V.A., Zanchurin M.A., Kozitsyn A.S., Krivchikov M.A., Shachnev D.A. *Arkhitekturno-tekhnologicheskie aspekty razrabotki i soprovozhdeniia bolshikh informatsionno-analiticheskikh sistem v sfere nauki i obrazovaniia* [Architectural and technological aspects of the development and maintenance of large information and analytical systems in the field of science and education]. *Programmnaia inzheneriia* = Software Engineering. 2017;8(10):448-455. (In Russ., abstract in Eng.) https://doi.org/10.17587/prin.8.448-455
[6] Vasenin V.A., Zenzinov A.A., Lunev K.V. *Ispolzovanie naukometricheskikh informatsionno-analiticheskikh sistem dlia avtomatizatsii provedeniia konkursnykh protsedur na primere informatsionno-analiticheskoi sistemy ISTINA* [The use of scientometric information and analytical systems to automate the conduct of competitive procedures on the example of the information and analytical system ISTINA]. *Programmnaia inzheneriia* = Software Engineering. 2005;7(10):472-480. (In Russ., abstract in Eng.) https://doi.org/10.17587/prin.7.472-480
[7] Afonin S.A., Kozitsyn A.S., Shachnev D.A. *Programmnye mekhanizmy agregatsii dannykh, osnovannye na ontologicheskom predstavlenii struktury reliatsionnoi bazy naukometricheskikh dannykh* [Software data aggregation mechanisms based on the ontological representation of the structure of the relational database of scientometric data]. *Programmnaia inzheneriia* = Software Engineering. 2016;7(9):408-413. (In Russ., abstract in Eng.) https://doi.org/10.17587/prin.7.408-413

[8] Kozitsyn A.S., Afonin S.A., Shachnev D.A. *Metod otsenki tematicheskoi blizosti nauchnykh zhurnalov* [A method for assessing the thematic proximity of scientific journals]. *Programmnaia inzheneriia* = Software Engineering. 2020;(6):335-341. (In Russ., abstract in Eng.) https://doi.org/10.17587/prin.11.335-341

[9] Kozitsyn A.S., Afonin S.A., Zenzinov A.A. *Algoritm opredeleniia perevodov statei s ispolzovaniem statisticheskikh dannykh* [Algorithm for determining translations of articles using statistical data]. *Russian Digital Libraries Journal.* 2018;21(6):494-505. (In Russ., abstract in Eng.) EDN: VVFJBM

[10] Kozitsyn A.S., Afonin S.A. *Algoritm razresheniia neodnoznachnosti imen avtorov v IAS ISTINA* [The algorithm for resolving the ambiguity of the names of authors in the information and analytical system ISTINA]. *Modern Information Technologies and IT-Education*. 2020;16(1):108-117. (In Russ., abstract in Eng.) https://doi.org/10.25559/SITITO.16.202001.108-117

[11] Zenziniov A.A., Zanchurin M.A., Afonin S.A. Acquisition and verification of scientometric data. Methods and software tools. *CEUR Workshop Proceedings*. 2019;2514:356-367. Available at: https://ceur-ws.org/Vol-2514/paper67.pdf (accessed 27.08.2023). (In Russ., abstract in Eng.)

[12] Tripathi V., Agarwal R., Pandey P., Ajith Kumar S.P. Security Concerns in Data Warehouses: Implementation and Analysis of Virtual Private Database. In: 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom). New Delhi, India: IEEE Computer Society; 2019. p. 117-120. Available at: https://ieeexplore.ieee.org/document/8991323 (accessed 27.08.2023).

[13] Chen H.-Z. Oracle HTML DB Application with Virtual Private Database. *System Simulation Technology*. 2006;2(4):244-248.

[14] Eskandarian S., Zaharia M. ObliDB: Oblivious Query Processing for Secure Databases. *Proceedings of the VLDB Endowment*. 2019;13(2):169-183. https://doi.org/10.14778/3364324.3364331

[15] Han Z., Hu H. ProDB: A memory-secure database using hardware enclave and practical oblivious RAM. *Information Systems*. 2021;96:101681. https://doi.org/10.1016/j.is.2020.101681

[16] Sandhu R.S., Coyne E.J., Feinstein H.L., Youman C.E. Role-Based Access Control Models. *Computer*. 1996;29(2):38-47. https://doi.org/10.1109/2.485845

[17] Afonin S. Ontology Models for Access Control Systems. In: 2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC). Vladivostok, Russia: IEEE Computer Society; 2018. p. 1-6. https://doi.org/10.1109/RPC.2018.8482178

[18] Jin X., Krishnan R., Sandhu R. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC. In: Cuppens-Boulahia N., Cuppens F., Garcia-Alfaro J. (eds.) Data and Applications Security and Privacy XXVI. DBSec 2012. *Lecture Notes in Computer Science*. Vol. 7371. Berlin, Heidelberg: Springer; 2012. p. 41-55. https://doi.org/10.1007/978-3-642-31540-4_4

[19] Vasenin V.A., Iavtushenko E.D. *Sredstva soprovozhdeniia protsessov razgranicheniia dostupa k bolshim naukometricheskim dannym s ispolzovaniem mekhanizmov vizualnogo predstavleniia* [Tools for supporting the processes of delimiting access to large scientometric data using visual representation mechanisms]. *Programmnaia inzheneriia* = Software Engineering. 2020;(3):131-141. (In Russ., abstract in Eng.) https://doi.org/10.17587/prin.11.131-141

[20] Servos D., Osborn S. L. Current Research and Open Problems in Attribute-Based Access Control. *ACM Computer Surveys.* 2017;49(4):65. https://doi.org/10.1145/3007204

[21] Spendolini S. Virtual Private Database. *Expert Oracle Application Express Security*. Apress, Berkeley, CA, 2013. p. 211-223. https://doi.org/10.1007/978-1-4302-4732-6_12

[22] Liu J., Li X., Wang Sh. What have we learnt from 10 years of fintech research? A scientometric analysis. *Technological Forecasting and Social Change.* 2020;155:120022. https://doi.org/10.1016/j.techfore.2020.120022

[23] Wu P., Wang Z., Wang H., et. al. Visualizing the emerging trends of biochar research and applications in 2019: a scientometric analysis and review. *Biochar.* 2020;(2):135-150. https://doi.org/10.1007/s42773-020-00055-1

[24] Afonin S.A., Gaspariants A.E. *Avtomaticheskoe postroenie funkcii ocenki kachestva v zadache razreshenija neodnoznachnosti imen avtorov nauchnyh publikacij* [Construction of Quality Func-tion for Scientific Papers Author Names Disambiguation Problem Using Supervised Learning Techniques]. *Programmnaia inzheneriia* = Software Engineering. 2015;(10):31-37. (In Russ., abstract in Eng.) EDN: UNEXBN

[25] Kozitsin A.S., Afonin S.A. *Razreshenie neodnoznachnostej pri opredelenii avtorov publikacii s ispol'zovanie grafov soavtorstva v bol'shih kollekcijah bibliograficheskih dannyh* [The Resolution of Ambiguities in the Identification of Authors of the Publication with the Use of Co-Authors' Graphs in Large Collections of Bibliograph-ic Data]. *Programmnaia inzheneriia* = Software Engineering. 2017;8(12):556-562. (In Russ., abstract in Eng.) https://doi.org/10.17587/prin.8.556-562

Submitted 27.08.2023; approved after reviewing 30.09.2023; accepted for publication 09.10.2023.

About the authors:

Alexander S. Kozitsin, Leading Researcher of the Institute of Mechanics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), Cand. Sci. (Phys.-Math.), **ORCID: https://orcid.org/0000-0002-8065-9061**, alexanderkz@mail.ru

Maxim A. Zanchurin, Junior Researcher of the Institute of Mechanics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), **ORCID: https://orcid.org/0009-0006-2195-0920**, maxim.zanchurin@gmail.com

All authors have read and approved the final manuscript.