УДК 004.04 **DOI: 10.18413/2518-1092-2021-6-2-0-5**

Наумов Р.К.[1]
Самылкин М.С.[1]
Копейкин М.В.[2]

СПОСОБЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ СРЕДСТВАМИ СУБД

[1)] Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики», Кронверкский пр., д. 49, г. Санкт-Петербург, 197101, Россия
[2)] Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский горный университет», 21-я лин. В.О., д. 2, Санкт-Петербург, 199106

e-mail: ruslan.naumow.dake@gmail.com, maksamylkin@gmail.com

Аннотация
Рост объема неструктурированных данных, генерируемых различными приложениями и сервисами, и принятие компаниями факта ценности таких данных привели к востребованности систем, способных анализировать большие объемы данных без участия человека. Удовлетворить данную потребность могут системы интеллектуального анализа данных, однако повышение эффективности таких систем является актуальной задачей. Несмотря на растущую популярность NoSQL решений, основными системами управления базами данных все еще являются реляционные СУБД. В статье особое внимание уделено тому, что современные РСУБД могут использоваться не только в качестве надежных хранилищ данных. В настоящее время первоочередной задачей развития РСУБД является интеграция в них интеллектуального анализа данных. Благодаря тому, что данные остаются в хранилище, система не тратит ресурсы на выгрузку анализируемого набора данных из базы данных и загрузку результатов анализа обратно. Данный подход повысит как скорость разработки, за счет использования сервисов, вшитых в СУБД, так и производительность всей системы. В статье рассматриваются основные задачи интеллектуального анализа данных и существующие алгоритмы их решения. Описываются основные методики внедрения интеллектуального анализа данных в СУБД. Особое внимание уделено подходу, в котором система анализа данных рассматривается как внутренний сервис СУБД. В работе представлены известные системы и библиотеки анализа данных, разработанные для РСУБД, а также варианты расширений языка запросов SQL.
Ключевые слова: интеллектуальный анализ данных, реляционные СУБД, кластеризация, поиск шаблонов, классификация.
Для цитирования: Наумов Р.К., Самылкин М.С., Копейкин М.В. Способы интеллектуального анализа данных средствами СУБД // Научный результат. Информационные технологии. – Т.6, №2, 2021. – С. 32-40. DOI: 10.18413/2518-1092-2021-6-2-0-5

Naumov R.K.[1]
Samylkin M.S.[1]
Kopeikin M.V.[2]

DATA MINING METHODS USING DBMS TOOLS

[1)] Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, 49 Kronverkskiy prospekt, St. Petersburg, 197101, Russia
[2)] Saint Petersburg Mining University, 2 21-st line V.O., St. Petersburg, 199106, Russia

e-mail: ruslan.naumow.dake@gmail.com, maksamylkin@gmail.com

Abstract
The growth in the volume of unstructured data generated by various applications and services, and the acceptance by companies of the fact that such data is valuable, have led to the demand for systems that can analyze large amounts of data without human intervention. Data mining systems

can satisfy this need, but improving the efficiency of such systems is an urgent task. Despite the growing popularity of NoSQL solutions, the main database management systems are still relational databases. In the article, special attention is paid to the fact that modern RDBMS can be used not only as reliable data stores. Currently, the primary task of RDBMS development is to integrate data mining into them. Due to the fact that the data remains in the storage, the system does not waste resources on unloading the analyzed data set from the database and loading the analysis results back. This approach will increase both the speed of development, due to the use of services embedded in the DBMS, and the performance of the entire system. The article discusses the main problems of data mining and the existing algorithms for solving them. The main methods of implementing data mining in a DBMS are described. Special attention is paid to the approach in which the data analysis system is considered as an internal DBMS service. The paper presents well-known data analysis systems and libraries developed for RDBMS, as well as variants of SQL query language extensions.
Keywords: data mining, relational DBMS, clustering, pattern mining, classification.
For citation: Naumov R.K., Samylkin M.S., Kopeikin M.V. Data mining methods using DBMS tools // Research result. Information technologies. – T.6, №2, 2021. – P. 32-40. DOI: 10.18413/2518-1092-2021-6-2-0-5

ВВЕДЕНИЕ

На сегодняшний день спрос на эффективные и мощные инструменты для работы с данными вызван многими факторами. К ним можно отнести потребность в беспрерывных и высоконагруженных системах для анализа данных. Такая потребность объясняется ростом числа приложений и сервисов, постоянно генерирующих большие объемы неструктурированных данных. Скорость прироста таких данных для одного ресурса может превышать 1 Тб в день. Большие данные могут включать в себя документы, электронную почту, информацию социальных сетей, аудио, видеофайлы, и т.д. [1] Важными факторами в анализе данных являются их объем и скорость накопления, однако, критичным является наличие эффективных методов для их обработки.

Под интеллектуальным анализом данных понимают совокупность алгоритмов, методов и программного обеспечения для обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия стратегически важных решений в различных сферах человеческой деятельности. [2]

Для извлечения данных, т.е. полезных знаний, из Больших данных используются ETL-системы, которые включают в себя процессы трансформации и очистки данных. Такие процессы приводят к образованию массивных хранилищ данных. Один из разработчиков СУБД Ingres и PostgreSQL пишет [3], что для решения данной проблемы необходимо пользоваться технологиями, предоставляемыми СУБД.

По данным авторитетного портала DB-engines.com [4], собирающего данные о реляционных и NoSQL СУБД, на апрель 2021 года самыми популярными системами являются Oracle, MySQL, Microsoft SQL Server, PostgreSQL и MongoDB (рис. 1).

Из этой статистики можно сделать вывод, что, несмотря на растущую популярность NoSQL решений, разработчики все еще отдают большее предпочтение реляционным базам данных.

На конгрессе [5] специалистами в сфере обработки и анализа данных был поднят вопрос генерирования большого объема данных в процессе цифровой трансформации общества. Основным решением данной проблемы стал полный контроль всех процессов от получения данных до извлечения из них полезных знаний.

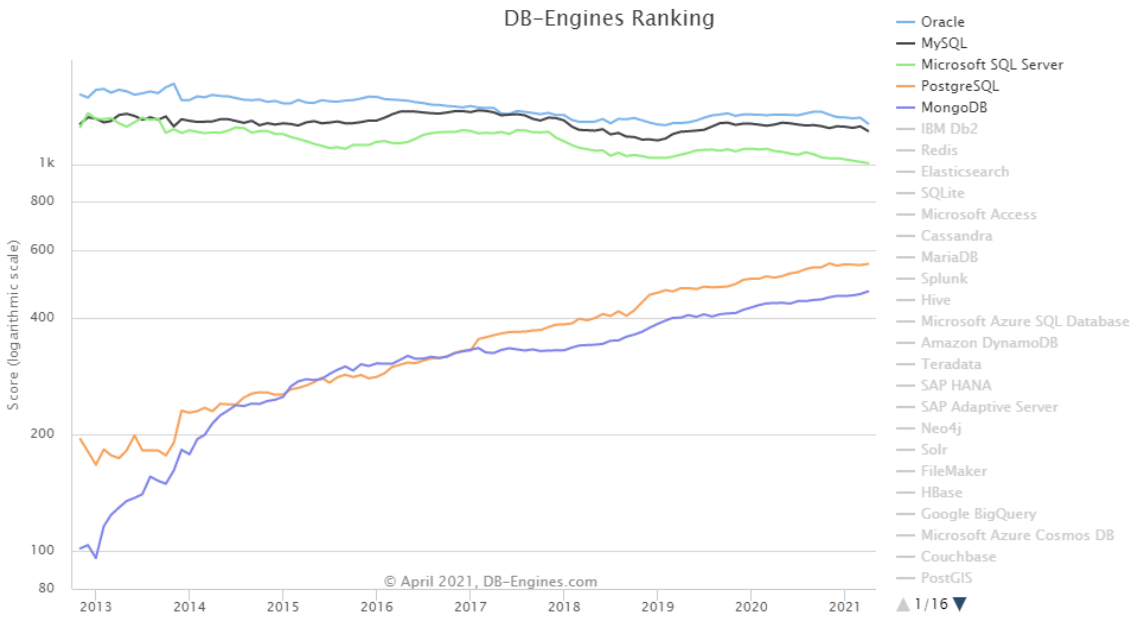

Рис. 1. Рейтинг СУБД портала DB-engines.com
Fig. 1. DB-Engines Ranking

На сегодняшний день внедрение методов интеллектуального анализа данных во внутренние процессы систем управления базами данных становится основной траекторией развития РСУБД. По расчетам автора работы [6] размещение верно подобранных алгоритмов обработки данных в непосредственной близости к анализируемым данным способствует увеличению производительности системы. Такая интеграция позволяет достичь минимизации расходов по выгрузке анализируемых данных из базы данных и загрузке результатов анализа обратно. Помимо этого, при обработке данных внутри хранилища, программист может воспользоваться внутренними сервисами СУБД, например, механизмом репликации, целостности и отказоустойчивости. Ускорить обработку данных можно с помощью индексации полей, алгоритмов оптимизации и других средств, которые заложены в архитектуру РСУБД.

Целью статьи является исследование способов интеграции интеллектуального анализа данных в реляционные СУБД. Основными задачами исследования являются:

1. Обзор существующих алгоритмов решения наиболее часто встречающихся задач интеллектуального анализа данных, таких как, кластеризация и поиск шаблонов;
2. Поиск подходов к внедрению интеллектуального анализа данных в РСУБД;
3. Определение и анализ существующих методов интеллектуального анализа данных в РСУБД.

ОСНОВНАЯ ЧАСТЬ

Задачи кластеризации и поиска шаблонов

Под задачей кластеризации подразумевается разбиение множества объектов, имеющих похожую структуру, на неопределенные заранее группы (кластеры) в зависимости от схожести их свойств. Задача кластеризации применяется повсеместно: сегментирование изображений, анализ социальных исследований и т.д. В алгоритмах кластеризации используются следующие статистические данные: количество точек в ячейке, а также результаты агрегатных функций (min, max, avg и т.д.).

Основные алгоритмы кластеризации перечислены ниже:

- Алгоритм четкой кластеризации;
- Алгоритм нечеткой кластеризации;
- Алгоритм раздельной (partitioning) кластеризации;

- Алгоритм k-средних (k-means);
- Алгоритмы k-medoids и PAM;
- Агломеративный иерархический алгоритм;
- Дивизимный иерархический алгоритм;
- Плотностная (density-based) кластеризация;
- Решетчатая (grid-based) кластеризация;

Примером агломеративного подхода кластеризации является алгоритм AGNES, в то время как для дивизимного подхода реализован алгоритм DIANA. Одним из примеров плотностной кластеризации является алгоритм DBSCAN, а для решетчатой кластеризации можно выделить алгоритм STING.

Задача поиска шаблонов заключается в нахождении явных закономерностей на имеющемся наборе объектов. Поиск шаблонов часто применяется в разнообразных областях человеческой деятельности, начиная с медицины, заканчивая анализом звуковых дорожек.

Традиционно для решения задачи поиска часто встречающихся зависимостей между данными применяют алгоритм Apriori. Суть действия данного алгоритма заключается в последовательном преобразовании наборов кандидатов в независимые множества с последующим отбором этих кандидатов по удовлетворяющему числу поддержки. Однако, у данного алгоритма есть недостаток: при большом значении таких наборов и низком пороговом значении поддержки появляются значительные расходы во времени. Исходя из этого были разработаны улучшенные алгоритмы Apriori: AprioriTid, DHP, DIC, Eclat, Partition.

Еще одним способом поиска шаблонов является алгоритм FP-Growth, который основывается на построении FP-дерева. В такой структуре данных наборы и значения их поддержки хранятся более компактно. Данный алгоритм также был доработан – появились алгоритмы OpportuneProject и AFOPT.

Подходы к интеграции интеллектуального анализа данных в РСУБД

Становление интеллектуального анализа данных, как отдельной дисциплины, дало начало научным изысканиям в области внедрения интеллектуального анализа данных в РСУБД, но на сегодняшний день данное направление не перестает быть актуальным. В научных трудах [7, 8] выделяют понятие «связывание». Данный термин подразумевает под собой интеграцию между интеллектуальным анализом данных и системой управления базами данных Понятия «связывание» принято делить на слабое, среднее и сильное связывание.

Слабое связывание представляет собой распределенную систему, в которой подсистема интеллектуального анализа данных существует автономно и не зависит от СУБД. Анализирующая система пользуется средствами СУБД для выгрузки данных из базы данных и обратной загрузки результатов анализа в базу данных. Данный подход используют такие open-source системы анализа данных как RapidMiner, Pentaho и т.д.

Принцип среднего связывания основывается на том, что система интеллектуального анализа данных имеет возможность выполнять примитивные операции, которые часто используются на стадии преданализа данных, с помощью средств СУБД. Такими операциями можно считать операции индексации, соединения отношений, а также исполнение агрегатных функций. Однако при данном подходе все еще подразумевается отделение анализирующей системы от системы управления базами данных. Здесь СУБД используется в качестве хранилища заранее вычисленных промежуточных результатов интеллектуального анализа, используемых наиболее часто.

Идея того, что система интеллектуального анализа данных закладывается в архитектуру СУБД и рассматривается как ее внутренний сервис, основывает принцип сильного связывания. Он заключается в использовании средств, обеспечивающих выполнение запросов на анализ данных на самом сервере базы данных. Оптимизация и выполнение функций анализа данных основывается на встроенных в СУБД сервисах и методах обработки запросов. Использование методики

сильного связывания способствует повышению скорости разработки и производительности эксплуатации всей информационной системы, однако в то же время является тяжело реализуемым.

Принцип сильного связывания системы интеллектуального анализа данных и СУБД может быть реализован с помощью двух подходов: интеллектуальный анализа данных, как сервис СУБД, и интеллектуальный анализа данных, как функция, написанная на языке SQL или его расширениях. В свою очередь каждый подход делят на различные способы реализации, которые схематично представлены на рисунке 2.

Внедренная в СУБД подсистема представляет собой механизм, поддерживающий сторонний язык анализа данных или расширяющий язык SQL добавлением необходимых для анализа функций, операторов и т.д.

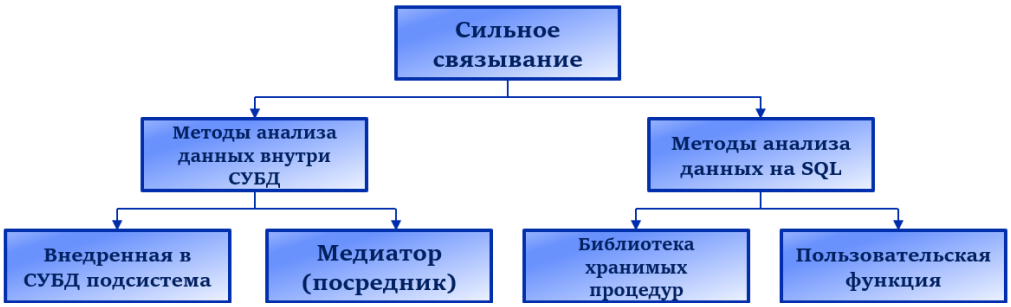

Рис. 2. Подходы к реализации «сильного связывания»
Fig. 2. Approaches to the implementation of "tight coupling" approach

Медиатор реализуется в виде посредника между архитектором баз данных и системой управления базами данных. Функцией медиатора является предоставление некоторого интерфейса или языка запросов для программиста, то есть обеспечивает преобразование запросов интеллектуального анализа данных в запросы на SQL.

Библиотека хранимых процедур разрабатывается программистом и представляет собой набор хранимых процедур, хранящихся в виде подпрограмм и компилирующихся однократно. После компиляция процедура постоянно хранится на сервере базы данных. При подключении библиотеки хранимых процедур к приложению базы данных процесс интеллектуального анализа выполняется внутри ядра СУБД.

Разработка пользовательских функций предполагает написание подпрограммы-функции, которая затем хранится на сервере СУБД. Пользовательская функция вызывается на исполнение с помощью вставки выражения в оператор SQL. Результат рассчитывается в процессе выполнения конкретного запроса и может быть представлен в виде скалярного, либо табличного типа. Пользовательская функция обычно реализовывается на SQL или его расширении, однако в большинстве современных СУБД возможна реализация на языках высокого уровня.

Методы интеллектуального анализа данных в РСУБД

Перейдем к обзору известных систем и библиотек для интеллектуального анализа данных, которые были разработаны для реляционных СУБД, а также вариантов расширений языка запросов SQL.

Занимающая одно из лидирующих мест среди инструментов для работы с базами данных СУБД Microsoft SQL Server (рис. 1) поддерживает стандарт OLE DB for Data Mining и специализированный язык запросов Data Mining Extensions (DMX). Язык DMX базируется на языке SQL, однако поддерживает только часть возможностей стандарта SQL:2009. Особенностью языка DMX является представление его операндов. Ими являются не традиционные реляционные таблицы (отношения), а сочетания данных, параметров, алгоритмов, применяющихся для анализа, и фильтров, задающих ход обработки данных. На рисунке 3 представлен пример запроса на языке DMX, а именно запрос кластеризации данных по заданным значениям.

```
SELECT PredHist(Cluster())
FROM [TM Clustreing]
    NATURAL PREDICTION JOIN
     (SELECT 1998 AS [Birth_Year],
   'Kingisepp' AS [City],
   1 AS [Childrens]) AS t
```

Рис. 3. Пример кластеризации данных на языке DMX
Fig. 3. Example of data clustering using the DMX language

Похожая реализация присутствует в СУБД Oracle, разработанной одноименной компанией, в виде модуля Oracle Data Mining [10]. При построении запроса к базе данных на интеллектуальный анализ данных используется PL/SQL API, реализованный пакетом DMBS_DATA_MINING. Пример запроса классификации данных на языке Oracle Data Mining представлен на рисунке 4. В первую очередь происходит создание модели "covid_risk", после чего происходит выборка по полученному в модели предсказанию (PREDICTION).

```
DBMS_DATA_MINING.CREATE_MODEL (
model_name => 'covid_risk_model',
    function => DBMS_DATA_MINING.classification,
data_table_name => 'covid_country_data',
case_id_column_name => 'country_id',
target_column_name => 'covid_risk',
    settings_table_name => 'credit_risk_model_settings');

SELECT country_name
FROM covid_country_data
WHERE PREDICTION (covid_risk_model USING *) = 'LOW'
```

Рис. 4. Пример классификации данных на языке Oracle Data Mining
Fig. 4. Example of data classification using the Oracle Data Mining language

Самым заметным средством интеллектуального анализа данных в реляционных СУБД PostgresSQL и разработанной на ее основе Greenplum является open-source библиотека MADlib. Широкий набор механизмов, предоставляемый данной подсистемой, дает возможность проводить кластеризацию и классификацию данных, осуществлять регрессионный анализ и пользоваться другими методами для анализа свойств данных. Особенностью библиотеки является адаптированность этих алгоритмов к реляционной составляющей системы без участия сторонних аналитических приложений. Обращение к базе данных происходит за счет исполнения заранее написанных на языке программирования Python пользовательских функций, которые выступают в роли коннектора и формируют корректную структуру таблиц. Пример вызова функции библиотеки MADlib, классифицирующую данные, представлен на рисунке 5.

```
SELECT madlib.create_nb_probs_view (
'example_feature_probs',          -- таблица выходных вероятностей
'example_priors',                 -- таблица выходных классов
'class_example_topredict',        -- таблица с данными для классификации
'id',                             -- имя ключевого столбца
'attributes',                     -- имя столбца атрибутов
3,                                -- количество атрибутов
'example_classified'              -- название нового представления (view)
);
```

Рис. 5. Пример классификации данных с помощью библиотеки MADlib
Fig. 5. Example of data classification using the MADlib library

Одним из вариантов расширения языка SQL для кластеризации данных является использование оператора CLUSTER BY, который был предложен в статье [11]. Работа оператора заключается в группировке результирующих строк с помощью встроенного алгоритма кластеризации. Данный механизм группировки отличается от традиционного, предусмотренного стандартом SQL, оператора GROUP BY, который выполняет группировку по точному совпадению значений в строках результирующей выборки. Одной из систем, в которой используется оператор CLUSTER BY, является PosgGIS – расширение СУБД PostgreSQL для работы с геоданными. Пример использования оператора приведен на рисунке 6. Существуют аналоги для различных СУБД, реализующие алгоритм кластеризации с помощью расширения языка SQL. К ним можно отнести SIMILAR GROUP BY для PostgreSQL, DISTRIBUTE BY для SPARK и др.

```
SELECT country_name
FROM counties
    WHERE mainland = 'Europe'
    CLUSTER BY population
```

Рис. 6. Пример кластеризации данных с помощью оператора CLUSTER BY
Fig. 6. Example of clustering data using the CLUSTER BY statement

Помимо приведенных ранее способов интеллектуального анализа данных существует исследовательское направление реализации алгоритмов анализа данных в РСУБД. Данные реализации позволят без дополнительных манипуляций над кодом переносить алгоритмы между различными СУБД. В таблице 1 приведены наиболее заметные в научных трудах SQL-реализации задачи поиска шаблонов.

Таблица 1

SQL-реализации задач поиска шаблонов

Table 1

SQL implementations of pattern mining tasks

Используемый алгоритм	**SQL-реализация**
Apriori	K-Way-Join
	Three-Way-Join
	Subquery
	Two-Group-Bys
	Set-oriented Apriori
	RDB-MINER
Universal quantification	Quiver
FP-Growth	Propad
	FP-TDG

Помимо представленных ранее задач, с помощью SQL также предлагается решать задачи классификации [12]. Классификация также, как и кластеризация, является задачей разделения конечного числа объектов на группы (классы), однако в отличии от задачи кластеризации имеет заранее определенную структуру и семантику классов. Одним из основных подходов к классификации является построение дерева решений. Несмотря на то, что данные, построенные на графах, имеют не реляционную природу, использование интеллектуального анализа таких данных в РСУБД является актуальным направлением. Так, например, в работе [13] был предложен алгоритм анализа структур деревьев с помощью SQL, а в статье [14] авторы описали алгоритм поиска полного подграфа, который основывается на применении средств РСУБД.

ЗАКЛЮЧЕНИЕ

На сегодняшний день наблюдается тенденция ускоренного роста объема данных. Неструктурированные данные генерируются различными сервисами и приложениями, которые являются частью жизни большинства людей. Все больше компаний уделяют внимание не только привлечению клиентов, но и сбору данных о них, а для их обработки без участия человека используют средства интеллектуального анализа данных. Реляционные СУБД по оценкам специалистов и мнению сообщества занимают лидирующую позицию среди инструментов управления данными. Перспективной траекторией развития РСУБД является внедрение в них средств интеллектуального анализа данных. Интеграция повысит скорость обработки данных, за счет минимизации ресурсных расходов по выгрузке анализируемых выборок из базы данных и загрузке результатов анализа обратно. Помимо этого, программист сможет воспользоваться внутренними сервисами СУБД, заложенными в ее архитектуру.

В статье были рассмотрены существующие алгоритмы решения задачи кластеризации и задачи поиска шаблонов. Определены основные подходы к интеграции интеллектуального анализа данных: подход слабого, среднего и сильного связывания. Наибольшее внимание уделено последнему подходу, более удобному с точки зрения прикладного программиста, но требующему для интеграции больших усилий. Приведены примеры такой интеграции, реализованной на основе библиотек хранимых процедур и пользовательских функций. Также в работе представлены некоторые примеры расширения языка SQL с помощью добавления специальных операторов, а также описаны известные SQL-реализации алгоритмов анализа данных.

Список литературы

1. Соловьев А.И. Хранение и Обработка Больших Данных // Тенденции Развития Науки и Образования. 2018. № 6 С. 47–51.
2. Цымблер М.Л. Обзор методов интеграции интеллектуального анализа данных в СУБД // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. № 2. С. 32–62.
3. Stonebraker M., Madden S., Dubey P. Intel “Big Data” Science and Technology Center Vision and Execution Plan. SIGMOD Record. 2013. № 1. С. 44–49.
4. DB-Engines Ranking, 2021 г. URL: https://db-engines.com/en/ranking (дата обращения: 03.04.2021).
5. Abadi D., Agrawal R., Ailamaki A. The Beckman Report on Database Research // Commun. ACM. 2016. № 2. С. 92–99.
6. Ordonez C. Statistical Model Computation with UDFs // IEEE Trans. Knowl. Data Eng. 2010. № 12. С. 1752–1765.
7. Han J., Kamber M. Data Mining: Concepts and Techniques // Morgan Kaufmann, 2006. С. 743.
8. Sarawagi S., Thomas S., Agrawal R. Integrating Mining with Relational Database Systems: Alternatives and Implications // ACM SIGMOD International Conference on Management of Data. 1998. С. 343–354.
9. Salal Y.K., Abdullaev S.M. Using of Data Mining Techniques to Predict of Student’s Performance in Industrial Institute of Al-Diwaniyah, Iraq // Bulletin of the South Ural State University Series: Computer Technologies, Automatic Control, Radio Electronics. 2019. № 19. С. 121-130.
10. Соболева А.Д., Сабинин О.Ю. Разработка метода композиции алгоритмов машинного обучения для решения задачи прогнозирования на примере технологии Oracle Data Mining // Theoretical & Applied Science. 2018. № 3. С. 147-154.
11. Sun P., Huang Y., Zhang C. Cluster-By: An Efficient Clustering Operator in Emergency Management Database Systems // Web-Age Information Management – WAIM. 2013. С. 152–164.
12. Li J., Cheng T., Zhao Z. High Efficient Classification Mining Method for Regional Large Data Features under Complex Attribute Environment // Academic Journal of Manufacturing Engineering. 2020. № 18(3). С. 113-119.
13. Padmanabhan S., Chakravarthy S. HDB-Subdue: A Scalable Approach to Graph Mining // Data Warehousing and Knowledge Discovery, 11th International Conference. 2009. С. 325–338.
14. Srihari S., Chandrashekar S., Parthasarathy S. A Framework for SQL Based Mining of Large Graphs on Relational Databases // Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference. 2010. С. 160–167.

15. Integration of Data Mining Techniques to PostgreSQL Database Manager System / Viloria A., Acuña, G.C., Alcázar F., D.J., Hernández-Palma, H., Fuentes, J.P., Rambal, E.P. // Procedia Computer Science. 2019. С. 575–580.

16. Аверьянова Е.В., Малышева Е.Ю. Алгоритмы интеллектуального анализа данных в Microsoft SQL Server // Вестник Поволжского государственного университета сервиса. Серия: Экономика. 2017. № 1. С. 115-120.

References

1. Soloviev A.I. Storage and Processing of Big Data // Trends in the Development of Science and Education. 2018. No. 6 pp. 47–51.

2. Zymbler M.L. Overview of Methods for Integrating Data Mining into DBMS // Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2019. No. 2. pp. 32–62.

3. Stonebraker M., Madden S., Dubey P. Intel “Big Data” Science and Technology Center Vision and Execution Plan. SIGMOD Record. 2013. No. 1. pp. 44–49.

4. DB-Engines Ranking, 2021. URL: https://db-engines.com/en/ranking (date of the request: 03.04.2021).

5. Abadi D., Agrawal R., Ailamaki A. The Beckman Report on Database Research // Commun. ACM. 2016. No. 2. pp. 92–99.

6. Ordonez C. Statistical Model Computation with UDFs // IEEE Trans. Knowl. Data Eng. 2010. No. 12. pp. 1752–1765.

7. Han J., Kamber M. Data Mining: Concepts and Techniques // Morgan Kaufmann, 2006. pp. 743.

8. Sarawagi S., Thomas S., Agrawal R. Integrating Mining with Relational Database Systems: Alternatives and Implications // ACM SIGMOD International Conference on Management of Data. 1998. pp. 343–354.

9. Salal Y.K., Abdullaev S.M. Using of Data Mining Techniques to Predict of Student’s Performance in Industrial Institute of Al-Diwaniyah, Iraq // Bulletin of the South Ural State University Series: Computer Technologies, Automatic Control, Radio Electronics. 2019. No. 19. pp. 121-130.

10. Soboleva A.D., Sabinin O.Yu. Development of a Method for Composition of Machine Learning Algorithms for Solving a Forecasting Problem using the Example of Oracle Data Mining Technology // Theoretical & Applied Science. 2018. No. 3. pp. 147-154.

11. Sun P., Huang Y., Zhang C. Cluster-By: An Efficient Clustering Operator in Emergency Management Database Systems // Web-Age Information Management – WAIM. 2013. pp. 152–164.

12. Li J., Cheng T., Zhao Z. High Efficient Classification Mining Method for Regional Large Data Features under Complex Attribute Environment // Academic Journal of Manufacturing Engineering. 2020. No. 18(3). pp. 113-119.

13. Padmanabhan S., Chakravarthy S. HDB-Subdue: A Scalable Approach to Graph Mining // Data Warehousing and Knowledge Discovery, 11th International Conference. 2009. pp. 325–338.

14. Srihari S., Chandrashekar S., Parthasarathy S. A Framework for SQL Based Mining of Large Graphs on Relational Databases // Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference. 2010. pp. 160–167.

15. Integration of Data Mining Techniques to PostgreSQL Database Manager System / Viloria A., Acuña, G.C., Alcázar F., D.J., Hernández-Palma, H., Fuentes, J.P., Rambal, E.P. // Procedia Computer Science. 2019. pp. 575–580.

16. Averyanova E.V., Malysheva E.Y. Data Mining Algorithms in Microsoft SQL Server // Bulletin of the Volga State University of Service. Series: Economics. 2017. No. 1. pp. 115-120.

Наумов Руслан Кириллович, инженер мегафакультета трансляционных информационных технологий, студент факультета инфокоммуникационных технологий
Самылкин Максим Сергеевич, студент факультета безопасности информационных технологий
Копейкин Михаил Васильевич, кандидат технических наук, доцент факультета фундаментальных и гуманитарных дисциплин, кафедра информационных систем и вычислительной техники

Naumov Ruslan Kirillovich, engineer, student, Faculty of Translational Information Technologies
Samylkin Maxim Sergeevich, student of the Faculty of Information Technology Security
Kopeikin Mikhail Vasilievich, Candidate of Technical Sciences, Associate Professor of the Faculty of Fundamental and Humanitarian Disciplines, Department of Information Systems and Computer Engineering