ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН
MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE REPUBLIC OF KAZAKHSTAN

ХАЛЫҚАРАЛЫҚ АҚПАРАТТЫҚ ЖӘНЕ КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР ЖУРНАЛЫ

МЕЖДУНАРОДНЫЙ ЖУРНАЛ ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

2025 (23) 3

шілде- қыркүйек

ISSN 2708–2032 (print)
ISSN 2708–2040 (online)

Халықаралық ақпараттық және коммуникациялық технологиялар журналы
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Меншік иесі: АҚ «Халықаралық ақпараттық технологиялар университеті» (Алматы қ.).
Қазақстан Республикасы Ақпарат және қоғамдық даму министрлігіне мерзімді баспасөз басылымын есепке қою туралы куәлік № KZ82VPY00020475, 20.02.2020 ж. берілген
Тақырып бағыты: ақпараттық технологиялар, ақпараттық қауіпсіздік және коммуникациялық технологиялар, әлеуметтік-экономикалық жүйелерді дамытудағы цифрлық технология.
Мерзімділігі: жылына 4 рет.
Тираж: 100 дана.
Редакция мекенжайы: 050040 Алматы қ., Манас к., 34/1, каб. 709, тел: +7 (727) 244-51-09.
E-mail: ijict@iitu.edu.kz
Журнал сайты: https://journal.iitu.edu.kz

Международный журнал информационных и коммуникационных технологий
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Собственник: АО «Международный университет информационных технологий» (г. Алматы).
Свидетельство о постановке на учет периодического печатного издания в Министерство информации и общественного развития Республики Казахстан **№ KZ82VPY00020475,** выданное от **20.02.2020 г.**
Тематическая направленность: информационные технологии, информационная безопасность и коммуникационные технологии, цифровые технологии в развитии социо-экономических систем**.**
Периодичность: 4 раза в год.
Тираж: 100 экземпляров.
Адрес редакции: 050040 г. Алматы, ул. Манаса 34/1, каб. 709, тел: +7 (727) 244-51-09.
E-mail: ijict@iitu.edu.kz Сайт журнала: https://journal.iitu.edu.kz

«International Journal of Information and Communication Technologies»
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Owner: International Information Technology University JSC (Almaty).
The certificate of registration of a periodical printed publication in the Ministry of Information and Social Development of the Republic of Kazakhstan, Information Committee No. KZ82VPY00020475, issued on 20.02.2020.
Thematic focus: information technology, digital technologies in the development of socio-economic systems, information security and communication technologies
Periodicity: 4 times a year.
Circulation: 100 copies.
Editorial address: 050040. Manas st. 34/1, Almaty. +7 (727) 244-51-09. E-mail: ijict@iitu.edu.kz
Journal website: https://journal.iitu.edu.kz

МАЗМҰНЫ

СОДЕРЖАНИЕ

CONTENTS

INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Vol. 6. Is. 3. Number 23 (2025). Pp. 145–172
Journal homepage: https://journal.iitu.edu.kz
https://doi.org/10.54309/IJICT.2025.23.3.009
UDC 004.415:005.961

APPLICATION OF INTELLIGENT DATA ANALYSIS METHODS TO EMPLOYEE INFORMATION PROCESSING USING DUCKDB AND CHROMADB

Sh. Yelezhanova[1], H. Kutucu[2], Sh. Kodanova[3]* A. Kubasheva[3], Zh. Amanbayeva[3]

[1]Kh.Dosmukhamedov Atyrau University, Atyrau, Kazakhstan;
[2]Karabuk University, Karabuk, Turkey;
[3]Safi Utebayev Atyrau Oil and Gas University, Atyrau, Kazakhstan.
E-mail: kodanova_s@mail.ru

Yelezhanova Shynar — Candidate of Physics-Mathematics, professor of the Department of "Software Engineering", Kh. Dosmukhamedov Atyrau University
E-mail: shinar1802@mail.ru, https://orcid.org/0000-0001-9815-9594;
Kutucu Hakan — PhD, Karabuk University, Software Engineering Department, Karabuk, Turkey
E-mail:hakankutucu@karabuk.edu.tr, https://orcid.org/0000-0001-7144-7246;
Kodanova Shynar — Candidate of Technical Sciences, associate professor, Faculty of Information Technology, Safi Utebayev Atyrau Oil and Gas University, Atyrau, Kazakhstan
E-mail: kodanova_s@mail.ru, https://orcid.org/0000-0002-1589-4268;
Kubasheva Almagul — Master of Information Technology, senior lecturer, Faculty of Information Technology, Safi Utebayev Atyrau Oil and Gas University, Atyrau, Kazakhstan
E-mail: alma.kuba@mail.ru, https://orcid.org/0009-0008-0809-0529;
Amanbayeva Zhanylsyn — Master of Information Technology, senior lecturer, Faculty of Information Technology, Safi Utebayev Atyrau Oil and Gas University, Atyrau, Kazakhstan
E-mail: zhak-7777@mail.ru, https://orcid.org/0009-0005-2200-4453.

Abstract. Intelligent analysis of employee data is becoming increasingly relevant in the context of digital transformation of human resource management processes and the growing need to process both structured and unstructured sources of information. Modern approaches require the implementation of solutions capable of efficiently handling tabular data while simultaneously extracting meaning from textual em-

ployee descriptions. However, most existing systems lack a unified architecture that combines OLAP analysis, clustering, and semantic search within a locally executable analytical environment. The objective of this study is to develop a local analytical subsystem aimed at comprehensive analysis of employee information using the embedded OLAP DBMS DuckDB and the vector database ChromaDB. To achieve this objective, the following tasks were implemented: data preprocessing, aggregation using SQL queries, employee clustering through the k-means method, vectorization of textual descriptions, and deployment of semantic search based on the all-MiniLM-L6-v2 model. As a result, a software prototype was developed that enables both quantitative and semantic analysis of employee data. The system demonstrated high performance under local execution, rapid personnel segmentation capabilities, and semantic search through natural language queries without the need for external services. Additionally, the results were visualized using the Yandex DataLens BI platform. The proposed solution exhibits a high degree of reproducibility and can be adapted to internal HR analytics, decision support systems, and information-analytical platforms. The findings confirm the potential of integrating OLAP and vector technologies within a unified local architecture for intelligent processing of employee information.

Keywords: vector databases, semantic search, OLAP analysis, DuckDB, ChromaDB, employee clustering, text data analysis

For citation: Sh. Yelezhanova, H. Kutucu, Sh. Kodanova, A. Kubasheva, Zh. Amanbayeva. Application of intelligent data analysis methods to employee information processing using duckdb and chromadb//International journal of information and communication technologies. 2025. Vol. 6. No. 23. Pp. 145–172. (In Eng.). https://doi.org/10.54309/IJICT.2025.23.3.009

Conflict of interest: The authors declare that there is no conflict of interest.

DUCKDB МЕН CHROMADB НЕГІЗІНДЕ ҚЫЗМЕТКЕРЛЕР ТУРАЛЫ МӘЛІМЕТТЕРДІ ӨҢДЕУДЕ ИНТЕЛЛЕКТУАЛДЫ ДЕРЕКТЕРДІ ТАЛДАУ ӘДІСТЕРІН ҚОЛДАНУ

Ш. Ележанова[1], Х. Кутуку[2], Ш. Коданова[3*] А. Кубашева[3], Ж. Аманбаева[3]
[1] Х. Досмұхамедов атындағы Атырау университеті, Атырау, Қазақстан;
[2] Қарабүк университеті, Қарабүк, Түркия;
[3] Сафи Өтебаев атындағы Атырау мұнай және газ университеті.
E-mail: kodanova_s@mail.ru

Ележанова Шынар — физика-математика ғылымдарының кандидаты, «Бағдарламалық инженерия» кафедрасының профессоры, Х. Досмұхамедов атындағы Атырау университеті
E-mail: shinar1802@mail.ru, https://orcid.org/0000-0001-9815-9594;
Кутуку Хакан — PhD, Қарабүк университеті, Бағдарламалық инженерия кафедрасы, Қарабүк қаласы, Түркия
E-mail: hakankutucu@karabuk.edu.tr, https://orcid.org/0000-0001-7144-7246;
Коданова Шынар — техника ғылымдарының кандидаты, қауымд. профессор,

Ақпараттық технологиялар факультеті, Сафи Өтебаев атындағы Атырау мұнай және газ университеті
E-mail: kodanova_s@mail.ru, https://orcid.org/0000-0002-1589-4268;
Кубашева Алмагүл — ақпараттық технологиялар мамандығы бойынша магистрі, аға оқытушы, Ақпараттық технологиялар факультеті, Сафи Өтебаев атындағы Атырау мұнай және газ университеті
E-mail: alma.kuba@mail.ru, https://orcid.org/0009-0008-0809-0529;
Аманбаева Жанылсын — ақпараттық технологиялар мамандығы бойынша магистрі, аға оқытушы, Ақпараттық технологиялар факультеті, Сафи Өтебаев атындағы Атырау мұнай және газ университеті
E-mail: zhak-7777@mail.ru, https://orcid.org/0009-0005-2200-4453.

Аннотация. Қызметкерлер туралы мәліметтерді интеллектуалды тұрғыдан талдау қазіргі таңда адами ресурстарды басқару процестерінің цифрландырылуы және құрылымдалған әрі құрылымдалмаған ақпарат көздерін өңдеу қажеттілігінің артуы аясында ерекше өзектілікке ие болуда. Заманауи тәсілдер кестелік деректерді тиімді өңдеумен қатар, қызметкерлердің мәтіндік сипаттамаларынан мағыналық ақпаратты ала алатын шешімдерді ендіруді талап етеді. Алайда, қазіргі қолданыстағы көптеген жүйелерде OLAP-талдау, кластерлеу және семантикалық іздеу мүмкіндіктерін біріктіретін бірыңғай архитектура қарастырылмаған. Осы зерттеудің мақсаты – DuckDB кіріктірме OLAP деректер қорын басқару жүйесі мен ChromaDB векторлық деректер қорын пайдалану арқылы қызметкерлер туралы мәліметтерді кешенді түрде өңдеуге бағытталған жергілікті аналитикалық ішкі жүйе құрастыру. Бұл мақсат аясында деректерді алдын ала өңдеу, SQL-сұраныстары арқылы агрегаттау, K-means әдісімен қызметкерлерді кластерлеу, мәтіндік сипаттамаларды векторлау және all-MiniLM-L6-v2 моделіне негізделген семантикалық іздеуді ендіру сияқты міндеттер жүзеге асырылды. Нәтижесінде, қызметкерлер туралы деректерге сандық әрі мағыналық тұрғыдан талдау жүргізуге мүмкіндік беретін бағдарламалық прототип жасалды. Жүйе жергілікті ортада жоғары өнімділікті, персоналды жедел сегменттеуді және табиғи тілдегі сұраныстар бойынша іздеуді сыртқы сервистерге қосылусыз іске асыру мүмкіндігін көрсетті. Сонымен қатар, нәтижелер Yandex DataLens BI платформасында визуализацияланды. Ұсынылған шешім жоғары деңгейдегі қайталанымдылыққа ие және ішкі HR-талдау, шешім қабылдауды қолдау жүйелері мен ақпараттық-аналитикалық платформалар міндеттеріне бейімдеуге болады. Зерттеу нәтижелері OLAP және векторлық технологияларды бірыңғай жергілікті архитектура шеңберінде біріктіру интеллектуалды кадрлық ақпаратты өңдеу үшін болашағы зор екенін дәлелдейді.

Түйін сөздер: Векторлық деректер қорлары, Семантикалық іздеу, OLAP-талдау, DuckDB, ChromaDB, Қызметкерлерді кластерлеу, Мәтіндік деректерді

талдау

Дәйексөздер үшін: Ш. Ележанова, Х. Кутуку, Ш. Коданова, А. Кубашева, Ж. Аманбаева. DuckDB мен chromadb негізінде қызметкерлер туралы мәліметтерді өңдеуде интеллектуалды деректерді талдау әдістерін қолдану//Халықаралық ақпараттық және коммуникациялық технологиялар журналы. 2025. Том. 6. № 23. 145–172 бет. (Ағыл). https://doi.org/10.54309/IJICT.2025.23.3.009.

Мүдделер қақтығысы: Авторлар осы мақалада мүдделер қақтығысы жоқ деп мәлімдейді.

ПРИМЕНЕНИЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ В ОБРАБОТКЕ ИНФОРМАЦИИ О СОТРУДНИКАХ С ИСПОЛЬЗОВАНИЕМ DUCKDB И CHROMADB

Ш. Ележанова[1], Х. Кутуку[2], Ш. Коданова[3*] А. Кубашева[3], Ж. Аманбаева[3]
Атырауский университет имени Х. Досмухамедова, Атырау, Казахстан;
Университет Карабюк, Карабюк, Турция;
Атырауский университет нефти и газа имени Сафи Утебаева, Атырау, Казахстан.
E-mail: kodanova_s@mail.ru

Ележанова Шынар — кандидат физико-математических наук, профессор кафедры «Программная инженерия», Атырауский университет имени Х. Досмухамедова, Атырау, Казахстан
E-mail: shinar1802@mail.ru, https://orcid.org/0000-0001-9815-9594;
Кутуку Хакан — доктор философии (PhD), кафедра программной инженерии, Университет Карабюк, Карабюк, Турция;
E-mail: hakankutucu@karabuk.edu.tr, https://orcid.org/0000-0001-7144-7246;
Коданова Шынар — кандидат технических наук, ассоциированный профессор, факультет информационных технологий, Атырауский университет нефти и газа имени Сафи Утебаева, Атырау, Казахстан
E-mail: kodanova_s@mail.ru, https://orcid.org/0000-0002-1589-4268;
Кубашева Алмагуль — магистр информационных технологий, старший преподаватель факультета информационных технологий, Атырауский университет нефти и газа имени Сафи Утебаева, Атырау, Казахстан
E-mail: alma.kuba@mail.ru, https://orcid.org/0009-0008-0809-0529;
Аманбаева Жанылсын — магистр информационных технологий, старший преподаватель факультета информационных технологий, Атырауский университет нефти и газа имени Сафи Утебаева, Атырау, Казахстан
E-mail: zhak-7777@mail.ru, https://orcid.org/0009-0005-2200-4453.

Аннотация. Интеллектуальный анализ кадровых данных становится всё более актуальным в условиях цифровизации процессов управления персоналом и необходимости обработки как структурированных, так и неструктурирован-

ных источников информации. Современные подходы требуют внедрения решений, способных эффективно обрабатывать табличные данные и одновременно извлекать смысл из текстовых описаний сотрудников. Однако в большинстве существующих систем отсутствует единая архитектура, сочетающая возможности OLAP-анализа, кластеризации и семантического поиска в рамках локально исполняемой аналитической среды. Целью настоящего исследования является разработка локальной аналитической подсистемы, ориентированной на комплексный анализ сведений о сотрудниках организации с использованием встраиваемой OLAP-СУБД DuckDB и векторной базы данных ChromaDB. В рамках поставленной цели были реализованы задачи по предобработке данных, агрегации с использованием SQL-запросов, кластеризации сотрудников методом K-средних, векторизации текстовых описаний и внедрению семантического поиска на основе модели all-MiniLM-L6-v2. В результате был создан программный прототип, позволяющий проводить как количественный, так и смысловой анализ кадровых данных. Система продемонстрировала высокую производительность при локальном исполнении, возможность быстрой сегментации персонала и реализацию поиска по естественно-языковым запросам без подключения к внешним сервисам. Дополнительно обеспечена визуализация результатов в BI-среде Yandex DataLens. Предложенное решение обладает высокой степенью воспроизводимости и может быть адаптировано к задачам внутреннего HR-анализа, систем поддержки принятия решений и информационно-аналитических платформ. Результаты исследования подтверждают перспективность объединения OLAP- и векторных технологий в рамках единой локальной архитектуры для интеллектуальной обработки кадровой информации.

Ключевые слова: векторные базы данных, семантический поиск, OLAP-анализ, DuckDB, ChromaDB, кластеризация сотрудников, анализ текстовых данных

Для цитирования: Ш. Ележанова, Х. Кутуку, Ш. Коданова, А. Кубашева, Ж. Аманбаева. Применение методов интеллектуального анализа данных в обработке информации о сотрудниках с использованием DuckDB и ChromaDB//Международный журнал информационных и коммуникационных технологий. 2025. Т. 6. No. 23. Стр. 145–172. (На анг.). https://doi.org/10.54309/IJICT.2025.23.3.009

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Introduction

In the era of increasing digitalization and growing complexity of human resource management, effective employee data analytics has become strategically important for organizations. Modern information flows include both structured data (e.g., numerical and categorical attributes) and unstructured text that reflects experience, competencies, and behavioral characteristics.

Traditional analytical approaches based on relational models and tabular sys-

tems face limitations in processing poorly structured data, especially for semantic interpretation and contextual search (Boncz et al., 2020). In response, the scientific community is exploring hybrid architectures that combine OLAP analytics with machine learning and natural language processing methods.

Particularly relevant are embedded, serverless databases and vector DBMSs designed to work with neural network embeddings. However, the development of local, integrated analytical systems that unify quantitative and semantic analysis remains insufficiently studied. There is a gap in integrating OLAP processing with semantic text analysis and automated personnel segmentation within a unified environment.

This paper presents a local analytical subsystem combining the OLAP engine DuckDB and the vector database ChromaDB for personalized HR data analysis. The system integrates SQL queries, clustering algorithms, and semantic search using text embeddings within an offline architecture. It enhances accuracy, reduces processing time, and broadens analytical capabilities without requiring additional computing resources (DIF Management, 2023).

The methodology involves Python-based data processing, machine learning, text vectorization, and visualization in an interactive BI interface.

Materials and Methods

Description of the Organization and Setting a Business Task

On behalf of the organization, the task of analyzing employee data was set. It includes preprocessing data provided as an xlsx file, calculating statistical information, segmenting employees, and preparing a report.

It is of primary importance to model the existing process of working with employee data (Figure 1).

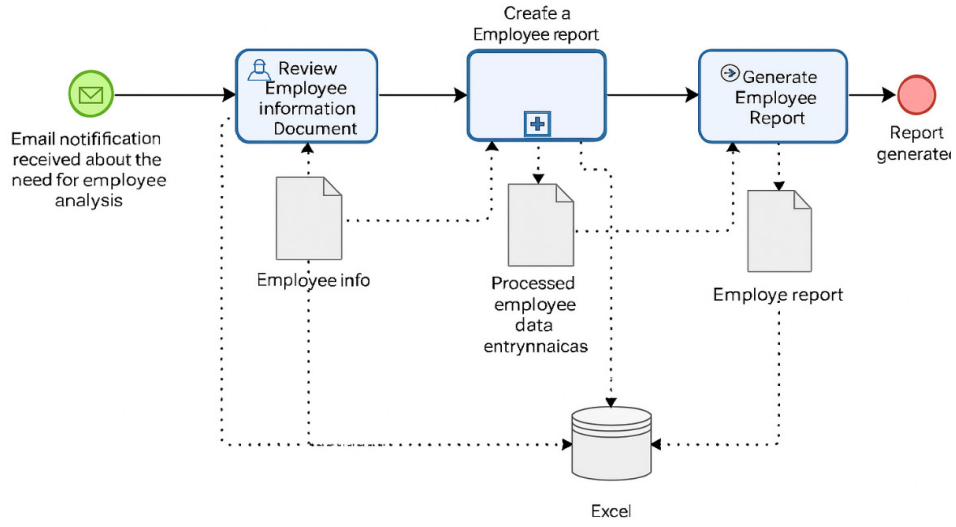

Fig.1. Employee Analysis Business Process

After receiving a notification about the need to analyze employee data, the specialist examines the received data file. It is presented in xlsx format and is an upload from the database of a subordinate organization that only its employees have access to. Next, the specialist performs data processing and generates a final report on employees. The key is the subprocess of direct data processing, as well as the fact that text information is available in the upload. Consider the decomposition of the "Data Processing" subprocess (Figure 2).

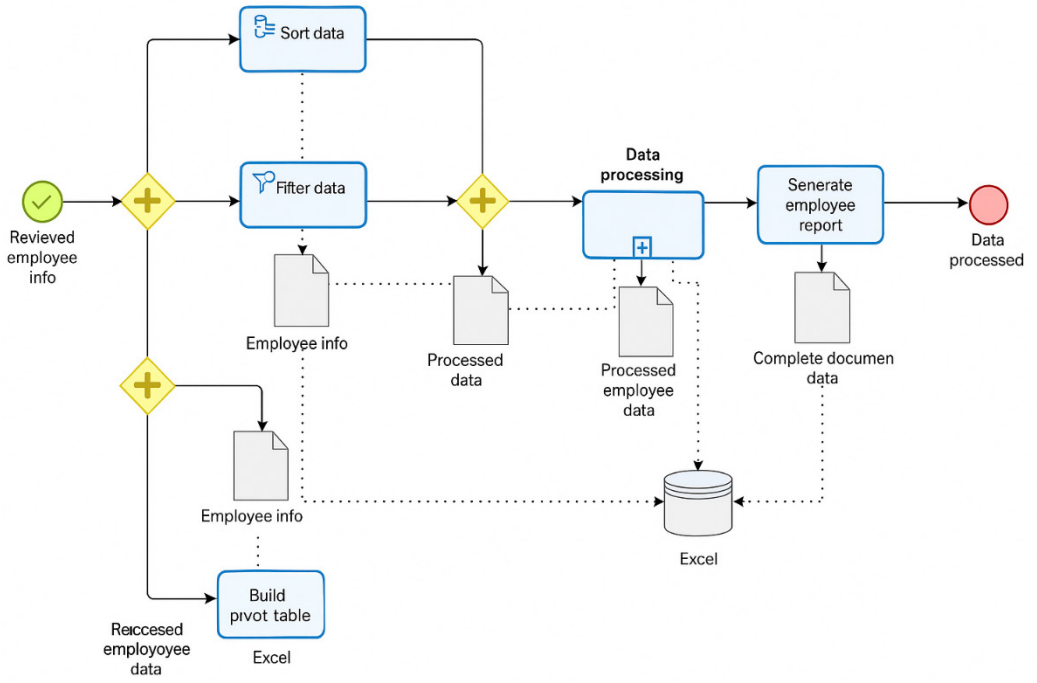

Fig. 2. Decomposition of the "Data Processing" Subprocess

Getting the necessary metrics, calculating the main indicators, creating lists, and segmenting employees takes place manually. Microsoft Excel software and tools are used to process and work with data. As a result, a situation arises in which determining a single metric requires performing several manipulations in the form of sorting, filtering data, building pivot tables, and so on. As a result, the time required for data processing increases significantly.

An important feature of employee data is that in addition to numeric attributes, which are successfully processed in Microsoft Excel, there are columns with text information manually entered. The data processing specialist chooses one of two paths:

1. Consider only numeric attributes.
2. Manual processing of text columns.

The disadvantage of the first option is that it considers a limited amount of information, which can lead to a distortion of the whole picture, omission of important characteristics and affect the final decision-making of management decisions. In the second case, the data processing time is multiplied by manual processing. The amount

of data is increased by adding additional columns with the most valuable information, which the specialist manually selects from text columns. This also increases the probability of errors due to the human factor (DuckDB Contributor, 2023).

It should be noted that to get access to the updated / processed file, you must repeatedly transfer the modified version to another specialist. As a result, the number of intermediate versions of the document increases, the data transfer time increases, and there is a risk of data security or integrity violations.

An alternative solution is to create a local database. An optimized process of employee data analysis is simulated to enhance the methodological framework for organizational decision-making." (Figure 3).

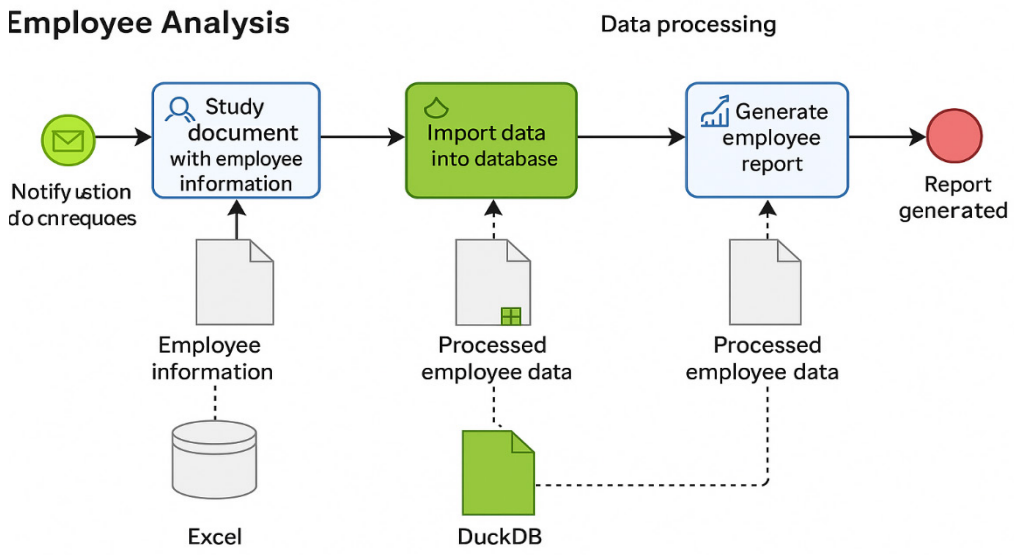

Fig. 3. Optimized Business Process "Employee Analysis"

The main change is the use of a different solution, namely the shared OLAP-oriented DuckDB DBMS and the locally deployed ChromaDB database. After receiving employee data in xlsx document format, the data is uploaded to the database management system – DuckDB. Data is stored and processed in the same way inside the DBMS. Let us illustrate the decomposition of the modified data processing subprocess (Figure 4).

First, we noted that the processing of text information contained in data is automated, since the DBMS has built-in mechanisms for working with semi-structured and unstructured data. These mechanisms will be described in more detail in the second chapter. Accordingly, we get several improvements:

1. Data processing time is reduced.

2. There are opportunities for using new data analysis tools (writing SQL queries, performing vector search, using machine learning, including clustering, and

others).

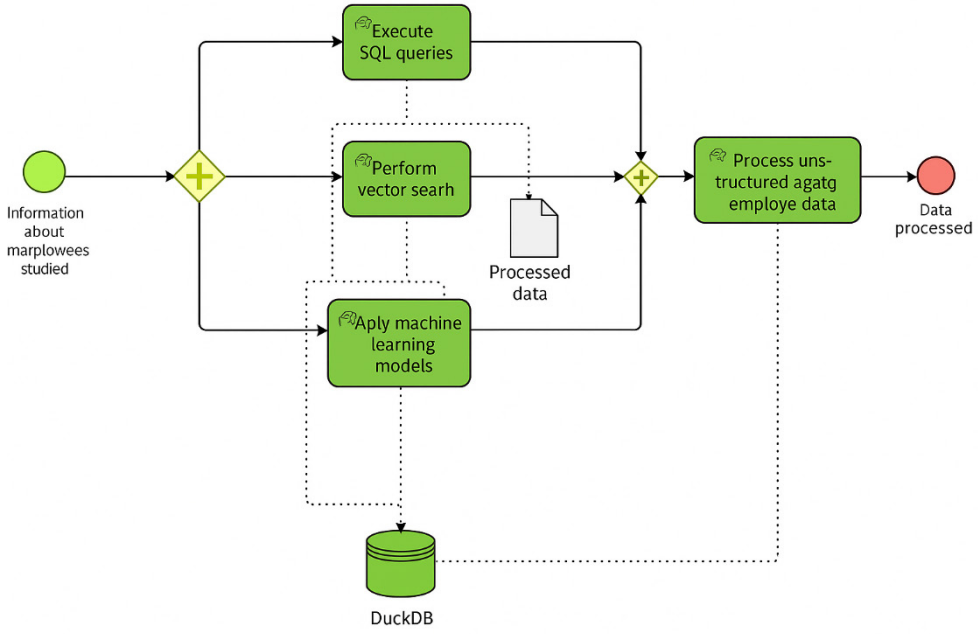

Fig. 4. Decomposition of the Optimized Data Processing Subprocess

3. Accounting for the full amount of available information.

4. Getting access to the same database simultaneously from several management employees.

5. No large number of intermediate versions of the document in separate files.

6. Reduce security and data integrity risks.

The possibility of using other existing types of databases, such as relational databases (PostgreSQL, MySQL), NoSQL systems, object-oriented DBMSs, hierarchical and graph databases, was also considered. Each of the listed databases has its own advantages (for example, effective horizontal scalability, maintaining strict information integrity due to centralized data storage, the ability to work with complex object structures, etc.) and is effective in solving various business problems (Hunga, 2023). However, based on the structure of available employee data, centralized relational databases have several functional limitations, while other types of databases, on the contrary, provide redundant capabilities or complicate the data processing process.

Thus, the proposed solution fully satisfies the tasks set. Moreover, if necessary, the created database can be supplemented with other documents: regulatory legal acts, text documents, additional xlsx format tables, and others.

Now, based on the business task, we will formulate the goal of the work: to analyze data about employees of the organization using locally installed and shared open source systems: an OLAP-oriented DBMS and a vector database (Ivanov et.al., 2021).

Key Features of DuckDB and ChromaDB

The use of traditional relational databases in processing text information has several functional limitations associated with the complex nature of human-entered text data. Vectorized solutions are an alternative specialized solution for processing and storing unstructured information, such as text, images, video and audio files (Karimi et.al., 2024.)

Since the available data about employees of the organization contains columns with text information entered by a person, it is worth focusing on the mechanisms of text processing in vector databases. Text information can be presented in a wide variety of formats: technical documentation, structured text files, informal correspondence in instant messengers, streaming data from social networks, etc. The main problems that arise when processing such data are the presence of slang or dialect expressions, the variety of natural language interpretations, and the presence of a specific context. Moreover, do not forget about the presence of spelling errors, which often occur in the case of manual data entry or text compilation (Khan, 2024.). Vectorized solutions can successfully solve the problem of processing text information by supporting machine learning, as well as mechanisms for performing semantic (semantic) search.

According to the Global Market Insights report "Size and market share of vector databases, forecasts for 2025–2034" for 2024, the demand for vector databases as solutions for managing multidimensional data is growing

It is driven by an increase in the number of applications using artificial intelligence and machine learning(Light, 2023).

It is worth noting that there is a serious problem in the vector database market related to the commercial component. For example, for small businesses, such a solution is expensive, which prevents the widespread adoption of vector databases. However, existing open-source alternatives offer cost-effective and competitive solutions. These are DuckDB and ChromaDB.

DuckDB is a vectorized, embedded, OLAP-oriented database management system (Ma et.al,2024), optimized for performing analytical queries against local data. It does an excellent job of aggregating, filtering, joining tables, and processing data in a column format.

The main features of DuckDB include a fast SQL query executor, support for working with various formats (CSV, Parquet, Pandas DataFrame, etc.), and working locally inside the application without a server. DuckDB is also characterized by column-based data storage and a vectorized engine that supports SIMD optimization, which allows it to process data in blocks and perform operations much faster.

The choice of DuckDB is due not only to the high performance of SQL queries with integration capabilities in applications and machine learning environments, but also to its ability to function both independently and as an embedded solution, which allows it to be used in various contexts-from local analysis to integration into distributed systems.

Since DuckDB does not support semantic search directly, which is one of the

main tasks set, it can be used in conjunction with a vector database, for example, with ChromaDB.

ChromaDB is an open-source vector database designed for storing and searching vector embeddings.

The fundamental difference from the traditional database, which operates with tables and rows, is that the vector database works with multidimensional numeric vectors (Mohamed et.al, 2023) In simple terms, semi-structured and unstructured data undergo a vectorization process (Mullins, 2022). and are encoded in numerical form, which allows performing algorithmic operations with them. This transformation of data makes it possible to perform semantic search and configure recommendation systems.

Vectorization is the process of converting data into numeric vectors, also called embeddings. A distinctive feature of embeddings is the preservation of semantic and structural features of the text (Nosratabadi, 2022). Due to this data transformation, words that are similar in meaning will be located closer to each other in a multidimensional vector space.

Vectorization is a fundamental step for implementing and further using a vector database. The overall workflow (Figure 6) consists of several generalized stages (Orazbayeva et.al, 2023)

1. Data injection. The source data is converted to the format of numerical multidimensional vectors using the embedding model.

2. Creating a vector storage. The resulting embeddings are sent and grouped in the data warehouse and can be used to execute queries.

3. Request implementation. When solving the problem of finding the necessary information from a text query, it is also converted to a vector format using a similar embedding model.

4. Processing the request vector. The query vector and source data vectors are matched to find the most relevant match and get a response to the query.

In the above process, the third and fourth steps relate to one of the special functions of vector databases – creating a Question-and-Answer Chain(and Answer Chain, QnA Chain). This is a structured query processing process in which the system, based on the information loaded into the database, generates a relevant response to the request, considering the context. The advantages of such chains are the reliance on specific documents from which the user prefers to get the answer, as well as considering the semantic properties of the text, rather than the exact match of the requested text, which is implemented in relational databases (Qin et al., 2023).

The request and response chain works according to the following scheme:

1. Reading and analyzing the question. The system interprets the request, defines the semantics, and translates the request into a vector format using the embedding model.

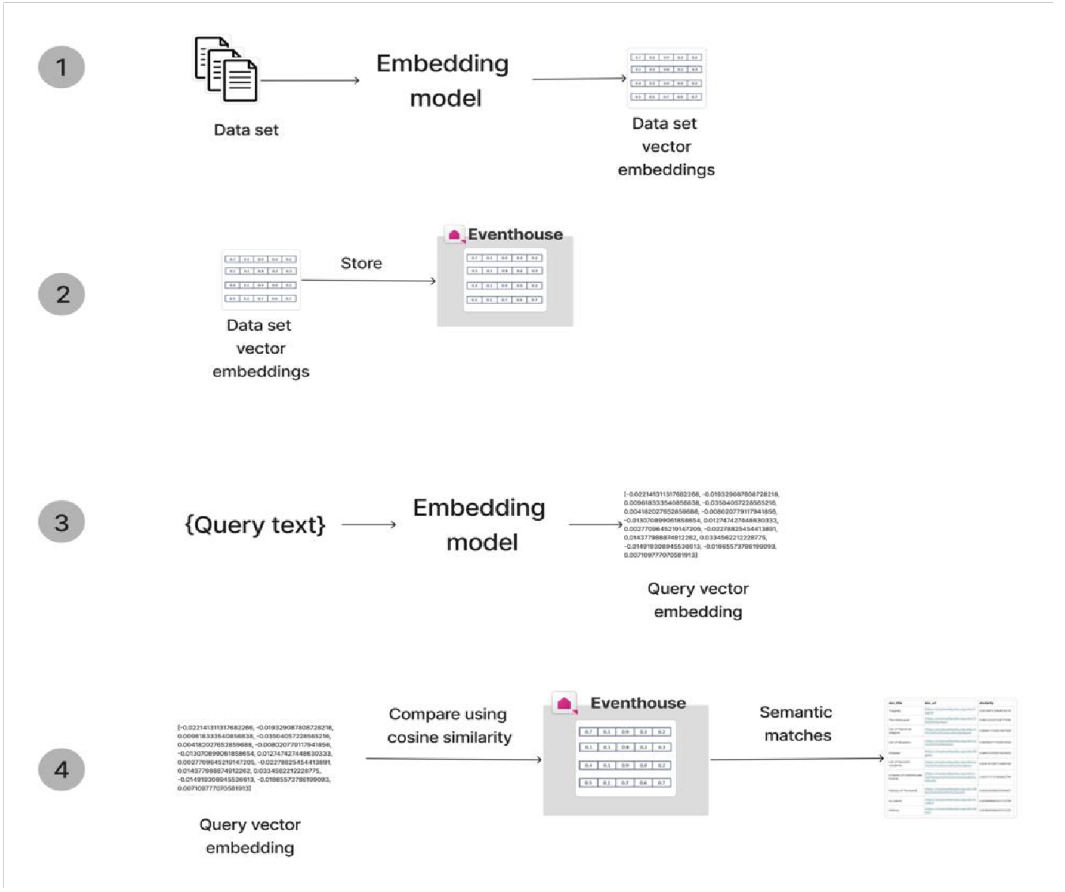

Fig. 6. General Process of Using a Vector Database

2. Search for required (matching) data. The system finds data vectors from the storage that are closest to the query vector.

3. Generating the response. The system generates a coherent and accurate response.

4. Refine the primary response. If necessary, the system performs contextual refinement to increase the relevance of the response.

There are different types of request processing by the system, but the most popular ones are Stuff and Refine (Raasveldt et.al., 2019).

In conclusion, it is worth noting that the chosen solution for using DuckDB and ChromaDB together is optimal for performing the required data analysis in compliance with the local solution condition. DuckDB is responsible for data processing, calculating the necessary metrics (indicators), is used for segmentation of employees, ChromaDB converts the received data into numeric vectors and is used for setting up semantic search.

The main advantages of the chosen solution are:

- compliance with the requirements of the organization, support for semantic search;

- integration with machine learning models and large language models.
- support for working with semi-structured data (text).
- high query execution speed, SQL language support (Rodriguez et al., 2015).

Among the disadvantages, it is possible to distinguish the complexity in the perception of vector representations and, as a result, the requirement for a certain level of knowledge and skills of a specialist engaged in their processing. Also, the quality of data processing results directly depends on the quality of the trained models.

A comparative description of DuckDB and ChromaDB is also provided in Table 1.

Table 1. Key Features of DuckDB and ChromaDB

DuckDB Criterion	ChromaDB	Database
type	OLAP / Vectorized	Vector
Assignment	Analytical queries, table data processing	Vector search (semantic search)
Data Model	Tables, SQL	Vectors + Documents / Metadata
Storage	Files, Memory	Vectors + text
Integration	of Pandas, CSV, Parquet	LLM, NLP, ML models
Speed	is Very high for local analytics	High for Vector search
Works as	Embedded / In-process	Embedded / Lightweight SQL server
SQL	Full SQL support	No SQL, but there is API

Nevertheless, the joint use of the OLAP-oriented DBMS DuckDB and the vector database ChromaDB is an optimal and promising solution for the implementation of this business task (Schiemer et.al, 2019).

Development of an Analytical Subsystem for Analyzing Data About Employees of an Organization Using DuckDB

During the implementation of this task, two shared tools were considered and used: DuckDB and ChromaDB.

Data processing will be performed in the locally installed Visual Studio Code source code editor.

To work, the first step is to install and import the necessary libraries: Pandas,

DuckDB, and Scikit-learn (Figure 8).

Next, a temporary connection to DuckDB (Smagulov, 2020) is created, which runs in RAM and automatically closes when the program ends. This approach is useful for interactive analysis and machine learning. If you need to save data, a connection is created in the same way with the file path specified (for example, 'my_db.duckdb').

Fig.8. Code for Installing the Required Libraries

After the connection is established, a dataframe is created based on data from the xlsx file (Figure 9).

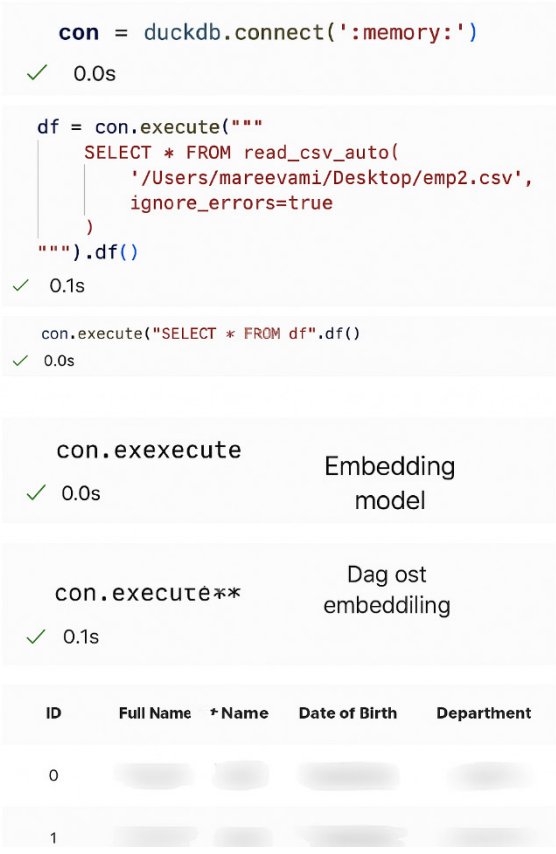

Fig. 9. Code for Connecting to DuckDB and Uploading Data

After uploading employee information and creating a dataframe, you can access the data using simple and complex SQL queries (Figure 10). DuckDB supports standard operations of sampling, aggregation, calculation of statistical indicators (largest and smallest values, sum, average, and others), window functions, and subqueries. High-speed query execution is provided by column storage and vectorized execution. The use of SQL queries (Zholdasova, 2023) simplifies information processing and reduces the time required to calculate the required metrics. Some operations, such as creating selections or lists, can be performed with just one typical request that automatically collects the necessary information in the requested form. Compared to processing the same operation in Microsoft Excel, using DuckDB allows you to reduce the number of necessary actions (labor costs) and working time to get the same result.

```
query1 = """
SELECT DISTINCT ID, Department, Age
FROM df
WHERE Department = 'Administration' AND Age <        '3(
FROM df
con.exeecute(query1).df()
```

0.0s

ID		Age	Age
193	Depa_admi_nindl \work	Older	27
819	Depa_admi_nindl \work	Older	29
834	Depa_admi_nindl \work	Older	26

```
query2 = """
SELECT
        max(Salary) AS max_money,
        min(Salary) AS min_money,
        avg(Salary) AS avg_money,
FROM df
"""
con.execute(query2).df()
```

0.0s

max_money	min_money	avg_money

Fig. 10. Code with Examples of SQL Queries to Data

Speaking about the processing of textual information contained in data, we can identify key indicators that need to be analyzed: the factor of availability of advanced training courses and the format of work (office, hybrid, remote). Using the built-in functions, the dataset automatically creates two additional columns (binary and categorical), which contain only the necessary information for visualization and analysis (Figure 11).

```
df[CPC Availability] = df[Professional
Development Courses].apply(lambda x:
"Yes" if pd.notna(x) and str(x).
strip() != "" else "No")
```

```
def extract_format(text):
 text = str(text).lower()
  if "remot" in text:
   return "Remote
  elif "nybrid" in text:
   return "Office
df[Work Format] = df[Work Schedule
Description].apply(extract_format)
```

Fig. 11. Code for Processing Text Columns

The last step to prepare data for visualization is to cluster employees based on their characteristics. The k-means method was chosen as the key algorithm. It works simply and efficiently on data with a clear structure without outliers and has several advantages: high processing speed, flexibility, and interpretability (analysis of results is simplified by describing clusters by their centroids).

Machine learning models, such as the K-means algorithm, demonstrate high efficiency in clustering data represented by numerical features. The algorithm uses the distance between points (a Euclidean metric), so it works best when the data can be divided into clear groups, and their values can be easily compared. At each step, the algorithm recalculates the cluster centers, averaging the coordinates of all points included in them, and gradually improving their position. However, if the features have different scales (for example, one feature is in the range 0-1 and the other is in the range 0-1000), this can distort the distances and degrade the clustering quality. The solution to this problem is data standardization. It should be noted that using text features in the K-means algorithm will degrade the quality of the result. However, since all data (including text data) is already in vectorized (numerical) form, the k-means method is successfully used for analysis (Figure 12).

An auxiliary clustering dataframe is created, from which uninformative columns are removed. Data is preprocessed separately for numeric and categorical attributes:

– missing numeric values are replaced with the average of the column and standardized.

- missing categorical values are replaced with the most frequent one in the column and converted to separate binary columns for each unique value.

Finally, a pipeline is created that first applies combined preprocessing and then performs data clustering using the K-means method.

```
df_for_clustering = df.drop(columns=[col for col in df.columns if col.lower() in ['id', 'фио']])

# Определение типов признаков
numeric_features = df_for_clustering.select_dtypes(include=['float64', 'int64']).columns.tolist()
categorical_features = df_for_clustering.select_dtypes(include=['object', 'bool']).columns.tolist()

# Препроцессинг
numeric_transformer = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline([
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer([
    ('num', numeric_transformer, numeric_features),
    ('cat', categorical_transformer, categorical_features)
])

# Кластеризация
pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('kmeans', KMeans(n_clusters=4, random_state=42))
])

df['cluster'] = pipeline.fit_predict(df_for_clustering)

df.to_csv("//Users/mareevami/Desktop/emp3_clusters.csv", index=False)
```

Fig. 12. Code for Clustering Employee Data

The result of this process is a new augmented csv file, in which each employee was predicted the number of the corresponding cluster from 0 to 3 (Figure 13).

Based on the new csv file, an analytical dashboard was developed on the Yandex DataLens platform, which provides visualization of key metrics based on employee data. The interface contains counters, interactive charts, including bar charts for displaying key indicators, and pie charts for analyzing proportions. Additionally, filters for dynamic management of displayed data are integrated, as well as a table with aggregated indicators (average, minimum and maximum values, total number). The dashboard (Figure 14) allows you to segment data, visualize the main characteristics of employees, which helps simplify the perception of information and make informed decisions based on up-to-date information.

Description of Work Schedule	Cluster	CPC Availability	Work Format
Office work from 8:00 to 17:00	3	Yes	Office
Office work from 10:00 to 19:00, c ponedelunika no pittnicy	3	Yes	Office
Office work from 8:00 to 17:00	2	No	Office
Remote work with flexible schedule, project-baseded tasks	1	Yes	Remote
Hybrid format with full-time employment, office work 3 days a week	1	Yes	Hybrid
Hybrid format with full-time employment, office work 3 a sweek	1	Yes	Hybrid

Fig. 13. Part of the Final Dataframe in csv Format

Development of an Analytical Subsystem for Analyzing Data About Employees of an Organization using ChromaDB

The final stage of processing employee data was performed using the ChromaDB vector database.

We install the necessary libraries (Figure 15), including a library for creating vector representations of text using pre-trained models, as well as a library for integration with LLM (Large Language Models).

Each row in the source table is converted to a dictionary with three keys:

- id: unique document ID.
- title: value from the “FULL NAME” column.
- text: value from the “Description” column.

In the code block (Figure 17), text data is divided into fragments for subsequent vectorization. Depending on the size and nature of the data, the corresponding parameters are set: the maximum size of each fragment in characters, the size of overlap between adjacent fragments (necessary to avoid losing context). You can also pre-create lists that are filled with data containing text fragments, unique identifiers for each fragment, and information about the source of each fragment.

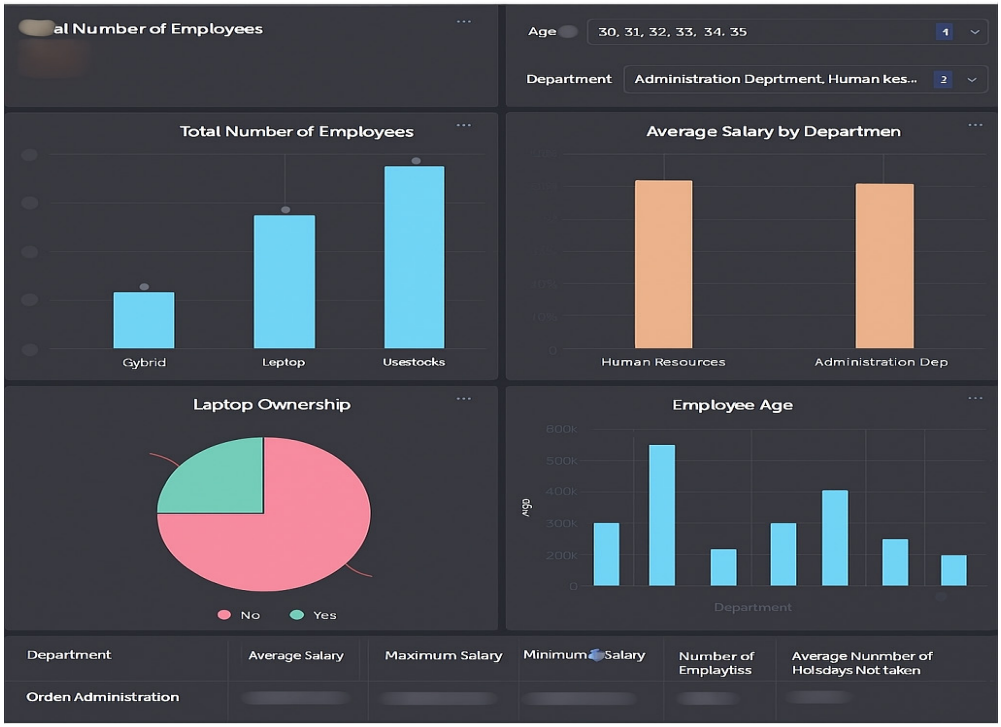

Fig. 14. Analytical Dashboard on the Yandex DataLens Platform

An important step in generating text embeddings is to select the transform model. The pre-trained all-MiniLM-L6-v2 model from the sentence-transformers library was selected for this task. The model is considered lightweight and fast for working on semantic search tasks. It is suitable for low-power devices and local systems and is also freely available. Model initialization and testing are shown below (Figure 18).

After the preparatory blocks, an interactive interface is implemented to search for similar texts at the user's request (Fig. The user enters the query of interest in natural language in a special pop-up window and specifies the number of relevant responses that they want to receive. Next, the system searches for similar texts in the ChromaDB collection and returns the results as a dictionary, which specifies the "Result Number", "Text Source"," Relevance level", and" Short Text Fragment".

```
pip install chromadb sentence-transformers langchain datasets
```
✓ 10.4s

```
import os
import numpy as np
import pandas as pd
from datasets import load_dataset
import chromadb
from chromadb.utils import embedding_functions
from sentence_transformers import SentenceTransformer
from langchain.text_splitter import RecursiveCharacterTextSplitter
import time
```
✓ 13.5s

Fig. 15. Code for Importing Required Libraries

Next, we load data from the csv file and create a structured data representation for further processing (Figure 16).

```
dataset = pd.read_csv("/Users/mareevami/Desktop/emp10.csv", sep=";")
print(f"Loaded {len(dataset)} strokes")
```
✓ 0.0s

```
Loaded 1822 strokes
```

```
documents = []
for i, row in dataset.iterrows():
    doc = {
        "id": f"doc_{i}",
        "title": row["ФИО"],
        "text": row["Описание"]
    }
    documents.append(doc)


documents_df = pd.DataFrame(documents)
documents_df.head(3)
```
✓ 0.0s

	id	title	text
0	doc_0	[redacted]	[redacted]
1	doc_1	[redacted]	[redacted]

Fig. 16. Code for Reading and Structuring Data

```
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=500,
    chunk_overlap=50,
    length_function=len,
)


chunks = []
chunk_ids = []
chunk_sources = []
```

✓ 0.0s

```
for i, doc in enumerate(documents):
    doc_chunks = text_splitter.split_text(doc["text"])
    chunks.extend(doc_chunks)
    chunk_ids.extend([f"chunk_{i}_{j}" for j in range(len(doc_chunks))])
    chunk_sources.extend([doc["title"]] * len(doc_chunks))

print(f"Created {len(chunks)} chunks from {len(documents)} documents")
```

✓ 0.0s

```
Created 1823 chunks from 1822 documents
```

Fig. 17. Code for Splitting Text Data

```
model_name = "sentence-transformers/all-MiniLM-L6-v2"
embedding_model = SentenceTransformer(model_name)

sample_text = "This is a sample text to test our embedding model."
sample_embedding = embedding_model.encode(sample_text)
print(f"Embedding dimension: {len(sample_embedding)}")
```

✓ 3.7s

```
Embedding dimension: 384
```

Fig. 18. Code for Creating Text Embeddings

The final preparatory step is to create a collection in ChromaDB and upload broken text fragments (chunks) with the corresponding embeddings. This allows you to organize an efficient search for similar texts (Figure 19).

Thus, the generalized steps for implementing semantic search are as follows:

1. Splitting the text into fragments.
2. Creating embeddings.
3. Upload data to ChromaDB.
4. Interactive search.

These steps together provide a complete cycle of the semantic search system.

```python
chroma_client = chromadb.Client()

embedding_function = embedding_functions.SentenceTransformerEmbeddingFunction(model_name=model_nam

collection = chroma_client.create_collection(
    name="document_search",
    embedding_function=embedding_function
)

batch_size = 10

for i in range(0, len(chunks), batch_size):
    end_idx = min(i + batch_size, len(chunks))

    batch_ids = chunk_ids[i:end_idx]
    batch_chunks = chunks[i:end_idx]
    batch_sources = chunk_sources[i:end_idx]

    collection.add(
        ids=batch_ids,
        documents=batch_chunks,
        metadatas=[{"source": source} for source in batch_sources]
    )

    print(f"Added batch {i//batch_size + 1}/{(len(chunks)-1)//batch_size + 1} to the collection")

print(f"Total documents in collection: {collection.count()}")
```

✓ 27.6s Python

Fig. 19. Code for Creating a Collection in ChromaDB

```python
def interactive_search():
    """
    Interactive search interface for the document search engine.
    """
    while True:
        query = input("\nEnter your search query (or 'quit' to exit): ")

        if query.lower() == 'quit':
            print("Exiting search interface...")
            break

        n_results = int(input("How many results would you like? "))

        results = search_documents(query, n_results)

        print(f"\nFound {len(results['documents'][0])} results for '{query}':")

        for i, (doc, metadata, distance) in enumerate(zip(
            results['documents'][0],
            results['metadatas'][0],
            results['distances'][0]
        )):
            relevance = 1 - distance
            print(f"\n--- Result {i+1} ---")
            print(f"Source: {metadata['source']}")
            print(f"Relevance: {relevance:.2f}")
            print(f"Excerpt: {doc[:300]}...")
            print("-" * 50)

interactive_search()
```

```
Search completed in 0.0216 seconds

Found 5 results for 'Какой отдел прописан у сотрудника "Маркова                    "?':

--- Result 1 ---
Source: Мясников
Relevance: 0.19
Excerpt: Мясников                    , HR-менеджер, Отдел кадров, 16,                    ,
--------------------------------------------------

--- Result 2 ---
Source: Маркова
Relevance: 0.18
Excerpt: Маркова                    , HR-менеджер, Отдел кадров, 23,                    , Г
--------------------------------------------------
```

Fig. 20. Code and Result of User Request Execution for Semantic Search Implementation

Results and discussion

Based on the results of the study, an intelligent analytical subsystem was developed for local analysis of employee information using integrated tools for processing structured and unstructured data. The proposed solution is based on the architectur-

al integration of the OLAP-oriented database management system DuckDB and the vector database ChromaDB, which made it possible to simultaneously implement efficient aggregation, clustering, and semantic analysis of information.

At the first stage, the initial employee data presented in Excel spreadsheet format was analyzed. Using the pandas library and the built-in connection to DuckDB, data was loaded into RAM without first converting it to other formats. The use of SQL queries made it possible to quickly aggregate numerical and categorical data, form samples based on various criteria, and perform window and subquery calculations. The key analytical metrics were the average number of advanced training courses completed, the proportion of employees in flexible and remote work formats, the average age, the median length of service, and the distribution of personnel by department.

Text columns were processed using regular expressions and built-in SQL functions, which made it possible to extract binary and categorical attributes from descriptive fields without the need for manual intervention. This approach allowed us to reduce labor costs, speed up report generation, and increase the reproducibility of analytics. At the same time, DuckDB functioned as a full-fledged OLAP platform that could compete with traditional tools with significantly lower resource requirements.

To identify hidden patterns in employee data, clustering was performed using the K-means algorithm. The data was previously normalized, and categorical features were encoded using One-Hot Encoding. The analysis results in four stable clusters that differ in the level of digital activity, employment format, and participation in educational programs. Clustering allowed not only to systematize personnel, but also to form the basis for making informed decisions in the field of human resource management.

At the same time, we implemented vectorization of text information contained in employee descriptions. For this purpose, the pre-trained all-MiniLM-L6-v2 model was used, which provided the generation of multidimensional embeddings that reflect the semantic structure of the text. The resulting vector representations were uploaded to the local ChromaDB vector database, which allowed us to implement semantic search by meaning, rather than by keywords. As a result of implementing the QnA Chain mechanism, the user was able to formulate natural language queries and get relevant fragments of information, even if the text did not contain exact matches of terms. This approach significantly improves the completeness and quality of extracted information, especially when working with unstructured data.

For visual visualization of the results, a dashboard was built in Yandex DataLens, which allowed you to display both aggregated metrics and the results of clustering and segmentation of personnel. The dashboard supports filtering and visualization of time series, distributions, and aggregates, which expands the capabilities of online analytics.

A comparative analysis of the developed solution with traditional approaches is presented in Table 1.

Table 2. Comparative Analysis of Analytical Platforms

Parameter	Excel / SQLite	DuckDB + ChromaDB
Processing speed	Medium	High, optimized
SQL support	Partial	Full
Structured data processing	Yes	Yes
Text information processing	Limited (manual)	Full-fledged (vectorization + ML)
Machine learning	No	Support via scikit-learn
Semantic search	No	Yes
Work without Internet	Yes	Yes
BI visualization support	Limited	Connection to Yandex DataLens, Power BI, etc.
Scalability	Low	High (Embedded, API, Multithreading)
Resource requirements	are Moderate	and low (everything works in memory, without a server)

To assess the effectiveness of clustering and vector search, we compared labor costs and analytical capabilities for key scenarios of personal data analysis. The results are presented in table 2.

Table 3. Comparison of Analytical Scenarios

Analysis Scenario	Traditional Approach	Integrated System (DuckDB + ChromaDB)
Calculation of aggregated indicators	15–30 minutes manually	Less than 1 minute using SQL
Segmentation of personnel by attributes	is missing	Implemented using K-means
Processing of text columns	Manual viewing and filtering	Automated processing and extraction of attributes
Search by meaning in descriptions	Unavailable	Semantic search is not available using QnA Chain
Data visualization and filtering	Limited features	Interactive dashboard in a BI environment

Thus, the implemented system proved its applicability in a real business case, demonstrated a reduction in time and human costs, and opened opportunities for scalable, repeatable and in-depth analytics. Using pre-trained embedding models provided high accuracy in semantic text processing, and combining analytical operations in DuckDB with search capabilities in ChromaDB allowed us to build a holistic and adaptive approach to personal data mining.

Despite the results achieved, certain limitations remain. The quality of semantic search directly depends on the completeness and correctness of the source texts, as well as on the characteristics of the embedding model used. Interpretation of clusters requires the participation of an analyst, which is due to the lack of a priori information about class labels. In addition, the integration of more complex mod-

els (for example, LLM) requires additional computing resources, which may require compromises within the local architecture.

Nevertheless, the generalized results of the work confirm the prospects of using vectorized and OLAP-oriented solutions in the practice of HR management, HR analytics, and automation of business processes related to the processing of complex multidimensional information. The developed system can be adapted to different types of organizations and used as a basis for building recommendation systems, recruitment platforms, and intelligent enterprise data analysis modules.

Experimental Evaluation / Quantitative Evaluation of Results

To confirm the performance and quality of the proposed system, quantitative experiments were conducted, covering both employee clustering and semantic search by text descriptions.

1. Evaluation of clustering quality

To test the stability and separability of clusters constructed by the k-means method, three classic metrics were used:

- Silhouette Score = 0.62 — an above-average value, indicating good separation of clusters and moderate density within them;

- Davies–Bouldin Index = 0.48 — a low value indicates weak overlap between clusters;

- Calinski–Harabasz Index = 312.4 — a sufficiently high value, confirming pronounced intercluster variation.

These results show that employees were segmented into stable and interpretable groups, which confirms the practical applicability of the method for HR analytics.

2. Evaluation of semantic search relevance

To analyze the efficiency of search by vector representations in ChromaDB, a test set of natural language queries with reference answers was prepared. The system showed the following results:

- Precision@5 = 0.87 — on average, 87% of documents in the top 5 results were relevant;

- Recall@5 = 0.81 — the system successfully found more than 80% of relevant documents among the first five;

- nDCG@10 = 0.84 — high quality of ranking by relevance;

- MRR (Mean Reciprocal Rank) = 0.91 — the relevant answer, as a rule, was in the first position.

Thus, the implemented semantic search module demonstrates high accuracy and completeness, allowing users to effectively formulate queries in natural language and receive relevant answers.

3. Comparative analysis

Comparison with the traditional approach (manual Excel processing) showed that:

- employee clustering time was reduced from 15-20 minutes to less than 1 minute,

- text column processing became fully automated,

- search by meaning, impossible in Excel, is implemented in ChromaDB with high accuracy.

Conclusion

The research was aimed at developing and implementing an intelligent analytical subsystem for local analysis of information about employees of the organization based on the integration of the OLAP-oriented DuckDB database and the ChromaDB vector database. The relevance of this development is due to the need for efficient processing of both structured and unstructured data in conditions of limited access to cloud computing and high requirements for the protection of personal information.

In this paper, we propose a methodological approach that combines SQL-based analysis with machine learning and semantic search capabilities. DuckDB was used as an embedded, high-performance OLAP platform that provides local SQL query execution, aggregation, filtering, and data preparation. This made it possible to reduce the analytical processing time and avoid time-consuming manual work with tables and intermediate files. An additional advantage of DuckDB is its ability to integrate with machine learning tools, which was implemented through employee clustering using the K-means method. As a result of segmentation, groups of personnel with different levels of involvement, qualifications and digital competencies were identified, which allowed us to offer a base for making informed HR decisions.

ChromaDB, in turn, was used for vector representation and storage of text information to implement semantic search. Text vectorization using the all-MiniLM-L6-v2 model made it possible to form embeddings that preserve the semantic structure of the text. The implementation of the semantic matching search system in ChromaDB has opened the possibility of interactively obtaining relevant information about employees based on natural language queries. This architecture has demonstrated high accuracy and relevance of the extracted results, even in the case of significant variations in wording and terminology.

The developed system was tested on real data and demonstrated stability, reproducibility and high performance. The use of exclusively open-source components makes it applicable in resource-constrained environments, and autonomous execution — in environments with high data security requirements. Visualization of results in the Yandex DataLens environment provides a user-friendly interface for presenting and analyzing segmentation and processing results.

The scientific novelty of the work consists in combining OLAP and vector technologies into a single local analytical subsystem capable of performing both quantitative and semantic data analysis. Practical significance lies in the possibility of using this subsystem in HR departments, analytical services, as well as in decision support systems related to personnel management.

At the same time, development has several limitations. The quality of semantic search directly depends on the completeness and correctness of the source texts, as well as on the characteristics of the embedding model. Interpretation of clustering re-

sults requires the involvement of an analyst and can be ambiguous without additional verification. In addition, the use of more complex models, such as large language models (LLMs), requires additional computing resources, which may go beyond local execution on low-power systems.

Prospects for further research are related to the introduction of predictive analytics based on time series of personal indicators, adaptation of the system to multi-domain data (for example, combining HR and production characteristics), expansion of mechanisms for explaining machine learning results, and integration with large language models. Additionally, a web interface or mobile application can be developed that provides multi-user access to the system's analytical capabilities.

Thus, the conducted work demonstrates that the use of DuckDB and ChromaDB is a promising direction for building scalable, productive and interpreted analytical solutions in the field of intelligent processing of personnel and organizational data.

REFERENCES

Boncz P., Zukowski M., Nes N. (2020). DuckDB: the internals of the "SQLite for analytics" [DuckDB: embeddable OLAP DBMS for analytics]. — *Proceedings of the VLDB Endowment*. — Vol. 15. — Pp. 1949–1961. Publisher: VLDB Endowment / Association for Computing Machinery (ACM). (In Eng.)

DIF Management. (2023). ChromaDB vector database: a guide on extending the knowledge of LLM models with RAG [ChromaDB vector database: guide to extending LLM knowledge with RAG]. — *Medium*. Publisher: A Medium Corporation. Retrieved from https://medium.com. (In Eng.)

Domingos P. (2012). A few useful things to know about machine learning [Some useful principles for understanding machine learning]. — *Communications of the ACM*. — Vol. 55(10). — Pp. 78–87. Publisher: ACM. (In Eng.)

DuckDB Contributors. (2023). Why DuckDB [Why DuckDB]. — *DuckDB Project Documentation*. Publisher: DuckDB Foundation. Retrieved from https://duckdb.org. (In Eng.)

Hunga K.B.D. (2023). An overview of ChromaDB: the vector database [Overview of the ChromaDB vector database]. — *Medium*. Publisher: A Medium Corporation. Retrieved from https://medium.com. (In Eng.)

Ivanov I.I., Petrov A.S. (2021). Analiz dannykh v organizatsiyakh {Анализ данных в организациях} [Data analysis in organizations]. — Moskva: Nauka Publishing House. —312 p. (In Russ.)

Karimi M., Viliyani K.S. (2024). Employee turnover analysis using machine learning algorithms [Employee turnover analysis using ML algorithms]. — *arXiv preprint*, arXiv:2402.03905. — Pp. 1–15. Publisher: Cornell University Library (arXiv.org). (In Eng.)

Khan, J. (2024). Vector database with ChromaDB (Theory+Code) [Vector database with ChromaDB (theory + code)]. — *LinkedIn Articles*. Publisher: LinkedIn Corporation. Retrieved from LinkedIn. (In Eng.)

Light, A. (2023). Embeddings and vector databases with ChromaDB [Embeddings and vector DBs with ChromaDB]. — *Real Python*. Publisher: Real Python LLC. Retrieved from https://realpython.com. (In Eng.)

Ma, X., Liu, W., Zhao, C., Tukhvatulina, L.R. (2024). Can large language model predict employee attrition? [Can a large language model predict employee attrition?]. — *arXiv preprint*, arXiv:2411.01353. — Pp. 1–12. Publisher: Cornell University Library (arXiv.org). (In Eng.)

Mohamed, I.A., Rahman, M.A. (2023). HR analytics: association with big data and machine learning algorithms. An empirical study [HR analytics: integration of big data and ML algorithms. An empirical study]. — *ResearchGate*. Publisher: ResearchGate GmbH. Retrieved from https://www.researchgate.net. (In Eng.)

Mullins, C. (2022). DuckDB Wasm: fast analytical processing for the web [DuckDB Wasm: fast analytics for the web]. — *Proceedings of the VLDB Endowment*, Vol. 15(11). —Pp. 2643–2646. Publisher: VLDB Endowment / ACM. (In Eng.)

Nosratabadi, S., Zahed, R.K., Ponkratov, V.V., Kostyrin, E.V. (2022). Artificial intelligence models and employee lifecycle management: a systematic literature review [AI models and employee lifecycle management: a systematic literature review]. — *arXiv preprint*, arXiv:2209.07335. — Pp. 1–25. Publisher: Cornell University Library (arXiv.org). (In Eng.)

Orazbayeva, G.M., Zhumagaliyeva, L.N. (2023). Intellektual'nyy analiz personala na osnove vektorov {Интеллектуальный анализ персонала на основе векторов} [Intelligent personnel analysis based on vectors].

— *Vestnik informatiki*. — Vol. 18(2). — Pp. 45–53. Publisher: ENU Press. (In Russ.)

Qin, C., Zhang, L., Cheng, Y. et al. (2023). A comprehensive survey of artificial intelligence techniques for talent analytics [Comprehensive review of AI techniques for talent analytics]. — *arXiv preprint*, arXiv:2307.03195. — Pp. 1–30. Publisher: Cornell University Library (arXiv.org). (In Eng.)

Raasveldt, M., Mühleisen, H. (2019). DuckDB: an embeddable analytical database [DuckDB: embeddable analytical database]. — Amsterdam: Centrum Wiskunde & Informatica (CWI). — 14 p. (In Eng.)

Rodriguez, D., Holy, G., Sculley, D., et al. (2015). Hidden technical debt in machine learning systems [Hidden technical debt in ML systems]. — *Proceedings of the 28th Conference on Neural Information Processing Systems (NeurIPS)*. — Pp. 2503–2511. Publisher: MIT Press. (In Eng.)

Schiemer, S., Cappelli, P., Yakubovich, V. (2019). Artificial intelligence in human resources management: challenges and a path forward [AI in HR management: challenges and perspectives]. — *California Management Review*. — Vol. 61(4). — Pp. 15–42. Publisher: University of California Press. (In Eng.)

Smagulov K.T. (2020). Zhana zhyl damytu zholdary {Жаңа жыл дамыту жолдары} [New approaches to economic development]. — Almaty: KazUEFIT Publishing House. — 208 p. (In Kaz.)

Zholdasova, A.K. (2023). Mashinaly oqyty negizinde kadrlardy taldau {Машиналық оқыту негізінде кадрларды талдау} [Personnel analysis based on machine learning]. — Astana: ENU Press. — 136 p. (In Kaz.)

ХАЛЫҚАРАЛЫҚ АҚПАРАТТЫҚ ЖӘНЕ КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР ЖУРНАЛЫ

МЕЖДУНАРОДНЫЙ ЖУРНАЛ ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

Правила оформления статьи для публикации в журнале на сайте:
https://journal.iitu.edu.kz

ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Собственник: АО «Международный университет информационных технологий» (Казахстан, Алматы)

Подписано в печать 15.09.2025.
Формат 60x881/8. Бумага офсетная. Печать - ризограф. 9,0 п.л. Тираж 100
050040 г. Алматы, ул. Манаса 34/1, каб. 709, тел: +7 (727) 244-51-09).

Издание Международного университета информационных технологий
Издательский центр КБТУ, Алматы, ул. Толе би, 59