N E W S
OF THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
SERIES OF GEOLOGY AND TECHNICAL SCIENCES
ISSN 2224-5278
Volume 6, Number 438 (2019), 12 – 21 **https://doi.org/10.32014/2019.2518-170X.151**

UDK 004:85; 004.89;004.93

G. T. Balakayeva[1], C. Phillips[2], D. K. Darkenbayev[1], M. Turdaliyev[1]

[1]Al-Farabi Kazakh National University, Almaty, Kazakhstan,
[2]University of Newcastle upon Tyne, Newcastle, Great Britain.
E-mail: gulnardtsa@gmail.com, chris.phillips@newcastle.ac.uk,
dauren.kadyrovich@gmail.com, t_medet@mail.ru

USING NoSQL FOR PROCESSING UNSTRUCTURED BIG DATA

Abstract. This paper provides an analysis of nowadays big data processing technologies. For processing unstructured large amount data, which is extremely in demand now (data in the form of video and audio files, animations, diagrams, etc.) authors used actual technologies based NoSQL. A comparative analysis of some NoSQL databases, which authors conducted and presented, showed that the choice of MongoDB is preferable, which was due to the simplicity and efficiency of working with this database. In authors opinion, after their researches, which are described in this article, it is now simpler and desirable to use an unstructured database for processing large amounts of data. In this article presents the results of the development database interfaces, development deployment diagrams, verifying the reliability and integration of data on NoSQL, creation of real Web application. While using NoSQL databases, especially MongoDB, can be to use only two tables with links to each other. In our opinion, this option is more convenient and understandable. Especially, when solving complex problems. It is this feature that will be applied by authors in the future to solve complex problems that require processing of large amount unstructured data.

Key words: processing Big Data, unstructured data, NoSQL, Web application.

Introduction. As an actual example of processing unstructured data, consider the features of creating online systems. Most creation tools are mainly based on universal data for each course. In practice, not every course is compatible with others. Some courses will require some additional features, for other courses they will be more compatible if the unnecessary functionality of the system is removed. In our research we will analyze these functionalities, select tools and create a web application [8].

System requirements are defined as follows:

– For current time planned about 400 users at the same time.

– Response time to user: up to 3 seconds.

– At the moment about 10-15 courses are planned.

– It planned about 200 MB of disk space with already installed courses. Without courses about 40 MB, pure assembly without courses.

– It is planned to write in fast programming languages (C# is selected, as it was familiar to me than other programming languages).

– For fast file uploads, asynchronous query execution in the database is used.

– It planned to write on ASP.NET MVC technology, which was able to recommend itself as a reliable and fast framework.

– Send mail by means of language tools, or if necessary use SMTP servers to send a notification to the mail user.

According existing systems, among which: Coursera, Moodle, Stepik. Coursera currently has a very large database of courses and cooperates with well-known universities and firms: NSU, Yandex, etc. There are many free courses on this resource, but there are also paid courses for interesting themes. In addition, the resource is not possible to publish its course. Since the platform only works with universities

by well-known companies. This is affected by the fact that some courses that the user needs can be paid. Moreover, the subscription itself in this platform is not free. A trial version given for 7 days, and then you will have to pay the course [1].

Next, we will to analyses platform Moodle. Moodle is a course management system (e learning), also known as a learning management system or virtual learning environment (English). It is an abbreviation of English. Modular Object-Oriented Dynamic Learning Environment. It is a free (distributed under the GNU GPL license) Web application that provides the ability to create sites for online learning. It is a free platform and a person, who wants to open his courses can easily open them and promote them. But, on the other hand, it can turn into expensive entertainment, since this platform is very cumbersome, and thus requires too much resource to create, for example, specialized training centers, it is worthwhile to open courses on this engine. Very good tool with so many possibilities. However, this makes it not convenient tool for creating courses for ordinary users [2].

One more platform Stepik. This platform is similar to Coursera. In difference, there are paid and free courses. In addition, there is an opportunity to publish own course. There is a choice: the course will be paid or free of charge. If you use the free version, your course will be available to everyone. Moreover, you can paint a course according to plan. To schedule lectures, tests, etc. A very good tool for each user, for creating and teaching one or some courses [3].

All the platforms listed above did not specialize in big data, the processing technologies of which will be discussed below.

Databases for unstructured large amount data: comparative analyses. Since the speed of data processing and presentation is important, some SQL databases were not able to provide such opportunities, NoSQL databases were invented. Since the application on demand to store a large amount of data, with a large amount of file - video, high quality images, and documents of all sorts, the MongoDB database was selected [4, 5]. In addition, 10 more databases were compared. Each system will be brie y reviewed, and will also be evaluated by heuristics:

1. Database type: In this part, the given database will be defined to which type of NoSQL database.
2. Supported programming languages: List of all supported languages, i.e. in which languages you can write the client application.
3. Scalability: All NoSQL databases have scalability to some extent. Not always scalability has a good side, in some bases it hurts the system rather than a positive impact.
4. Ease of use: This will consider the ability of a quick start, or just a threshold entry.
5. System Type: Commercial or Free. This means that the system is paid or free. Commercial receives a score of 3, for the complexity of the system.
6. The intensity of support. With the free system, the update frequency and the last update date are taken into account. For closed systems, this part will be heavy.
7. Quality of use: it is understood how many users, as well as how many were downloaded this system.
8. Possibility of modification. With open systems, this property is positive [6, 7].

Table 1 – First part of databases

Criteria/Database	Hbase	Redis	CouchDB	Cassandra	Amazon Dynamo
Database type	0	0	0	0	0
Supported programming languages	2	0	1	0	1
Scalability	0	2	2	3	0
Ease of use	2	1	0	0	2
System Type	2	0	0	3	4
The intensity of support	0	0	0	3	0
Quality of use	3	3	3	0	3
Possibility of modification	2	0	3	3	4

Table 2 – Second part of databases

Criteria/Database	MongoDB	Big Table	Neo4j	Oracle NoSQL	Couchbase
Database type	0	0	0	0	0
Supported programming languages	1	3	1	3	2
Scalability	0	0	0	0	3
Ease of use	0	0	2	2	1
System Type	2	4	0	2	2
The intensity of support	0	0	0	0	1
Quality of use	3	3	3	3	3
Possibility of modification	2	4	2	2	2

Online learning System functionalities. Users of the system are the following entities: Administrator (Manager), Teacher and User. The administrator is given changes to the user data. By changing the data we mean the following concepts: Password changes (password reset), mail changes, user name changes, and distribution of roles. And also changes in the courses.

The administrator can change course data:

– change the name of the course.

– change the category of the course

– delete course

– create a course

- Download lesson

- Creating lessons for the course

- Creating a task for lessons

– Administrators can register for courses. Change the name of the course.

Teachers can:

– Create a course

– Creating lessons for the course

– Creating a task for lessons

– Download lesson lessons

– Delete your course

– Change the name of your course

– Change the description of your course

– Changing the lesson data

– Teachers can also sign up for courses.

Users can only search for courses and sign up for courses. All users can log in to the system, then they get their role.

The application will have the roles of user, teacher, and administrator. By the next UML diagram, we can see all the roles of the application, as well as their privileges, (functions that each of the roles can do).

Enroll to online courses. Here enroll to courses is a function with which each user can choose his course and enroll, and also pass a free course with control questions at the end. The structure of the database will have a representation (divided into components), which can be seen in the figure below. Sign In the functional responsible for user authentication. Authentication will use the HTTP authentication protocol, specifically the Digest protocol. Digest is a challenge-response scheme in which the server sends a unique nonce value, and the browser passes the MD5 hash of the user's password calculated using the specified nonce. A safer alternative to the Basic scheme for unprotected connections, but is subject to man-in-the-middle attacks (with the replacement of the scheme for basic). In addition, the use of this scheme does not allow the use of modern hash functions to store user passwords on the server. Manage users function provides the administrator with user data management, including data changes. If you lose the login data, the administrator could change the data for further successful user authentication in the system. Manage Courses function allows the administrator and the teacher to change the data about their courses, as well as delete them, if they are irrelevant. For the administrator, and the teacher will have

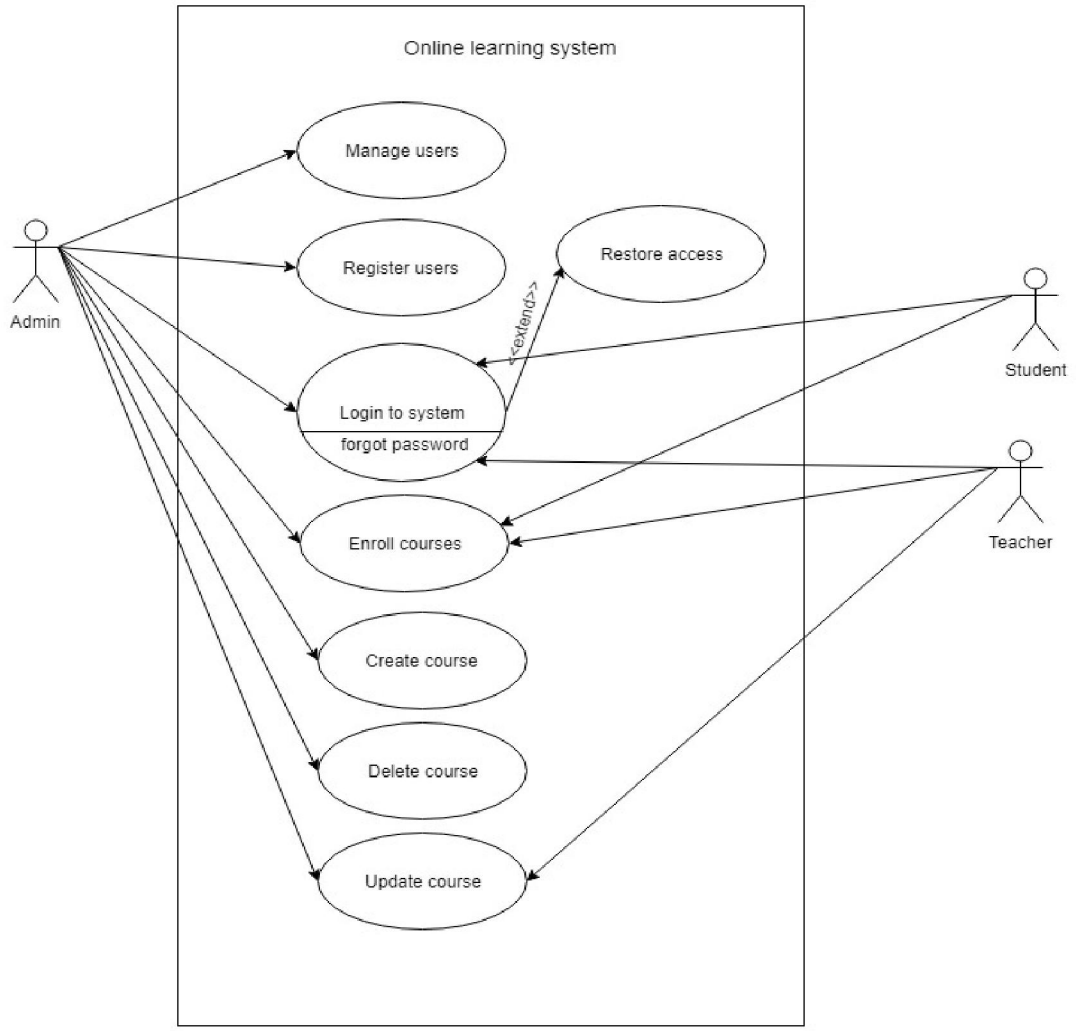

Figure 1 – Use-Case Diagram of users

different access rights to the course changes. The administrator is given the right to change for all courses, but for the teacher is given only for their courses, and can not be deleted and added. To do this, the teacher must ask the administrator for the given action.

The functionality enroll courses allows all users to enroll in courses, if the course is not protected by a secret word or was closed at the time of user recording, then the user can not enroll in the course. For these types of courses, the user should ask the teacher to give him access to this course.

The function of search courses provides users with the search for courses on the user's request. To do this, a simple search in the database will be used. If you match the search word or when you and any word from the user's query, it matches the word inside the names, description or tags when you use them. Also, there is a functionality for notifying users when answering a discussion or a user question. When answering a user's question, an email will be sent with the content of the answer to the question, with a link to this discussion with the answer. To use this functionality, SMTP, POP3 protocols must be enabled. In the absence of such functionalities, it is proposed to use existing SMTP services like Google.com, Mail.ru, Outlook.com. To facilitate the work with this functionality, the Administrator will need to select one of the above selected services, and enter the login and password from the user of this system. This data will be stored in the system itself and will not send spam. To disable the reply to the mail, there will be instructions for the issued SMTP service systems.

Online learning System Interface. Users will first see the main page of the courses, in which the menu presented, in which the possibility of registration or login provided. On the main page, users can view data about the project, as well as data about the team developing the system and teachers. In the

courses tab there will be a list of courses available in the database. On the main page you can immediately search for courses. Which will implement as a screenshot is given below. The list of all courses will also look like. In this screenshot there is no header, the same header will be used, for all pages from the main page.

Users can enter the system, then the main page for finding courses or viewing all available courses in the database will also be presented.

At the moment, the pages of the Main Page, the entry and registration page, and displaying the search result or displaying a list of all courses have been developed.

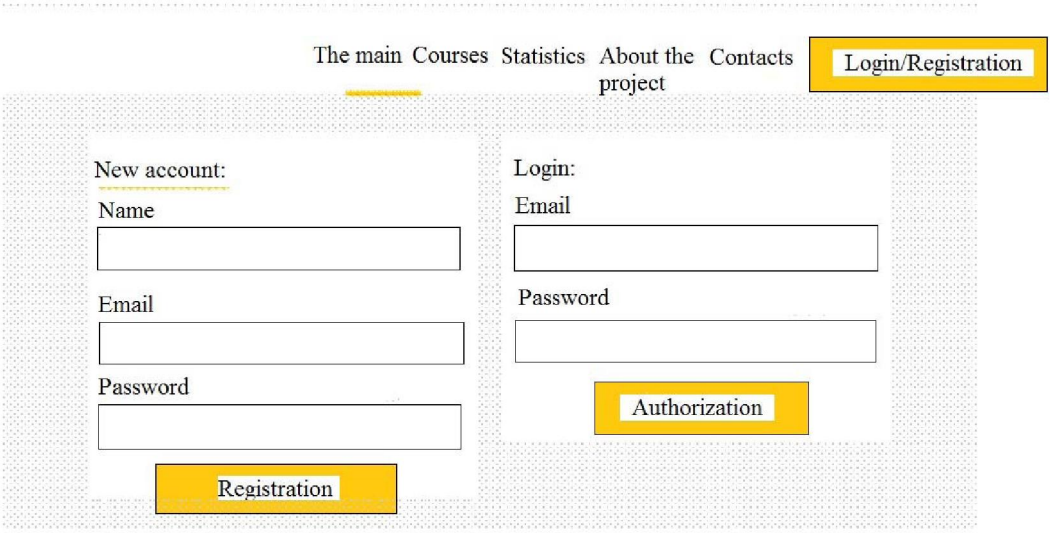

Figure 2 – Login and registration page

Web-application Architecture. Enroll Courses component provides functionality for working with courses, in cluding viewing and writing for courses. The Registration components respond by their name, for registration and authorization on the site. These components will be executed on the client side of the web-application.

In the Server section of the web application there will be such components as User Management-for the operation of managing users of the system, for authorizing and modifying user data by the administrator or by the user himself. It will also manage the definition of the user's role in authorization in the system.

In the database, there will be components for managing the databases that will contain all the queries associated with for modification, input, sampling (with or without criteria) or deleting data from the database.

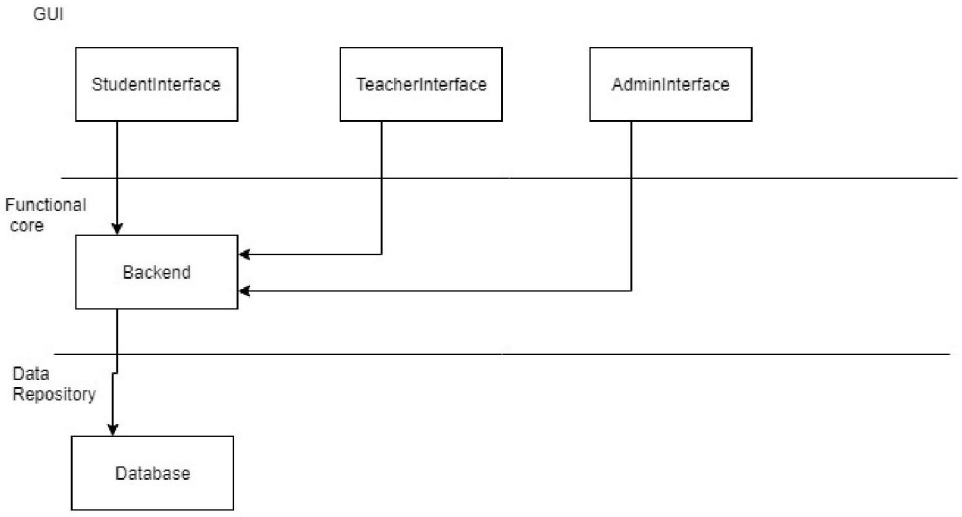

Figure 3 – System architecture

Database structure for Online course system's.

Database Entities. The following entities are distinguished in the database: Users, Courses, User Type, Lessons, Tasks, Documents and Chunks. Users entity provides users with information about users: name, mail, and password and user type. Courses are stored in essence Courses. In essence, data about the course is stored: name, description, tags. In essence, Lessons the data connected with a binding to courses is stored. It consists of the attributes: the course number, the document number and the content of the lesson.

The entities of the Documents and Chunks are linked together and retain documents relevant to the lessons. In essence, Chunks stores parts of documents, and the documents themselves are stored in the Documents entity.

In Essence Tasks - the data for the task is stored, test exercises for fixing the lesson. Data in this entity is closely related to data in the essence of Lesson.

The essence of the Forum, provides an answer, discussion, a given lesson, or an exercise relevant to this lesson. In essence, there will be attributes of the user ID, the identifier of the tied lesson, the user message.

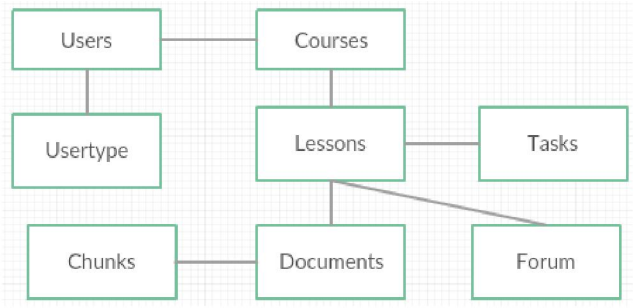

Figure 4 – Database entities

The figure provides an approximate view of the class diagram as the database will look.

Database structure. Entities are selected. Figure 5 diagram's shows relationships in a non-relational database, because the Users and Courses entities interact with each other in a many-to-many relationship (n: m). In relational data, you had to create another table, in which the data of the key fields of the two tables are stored. But with a non-relational database, you can solve this problem without an auxiliary table, simply by creating links between them. To preserve the integrity of the data, this method can also be used. Since when searching for data, they will search by ID in each table. After finding the correspondence of the data, the user will be given the data that is stored in the database. In relational databases, JOIN is used for such purposes. As distinct from the relational database, using some libraries, the Document and Chunks tables are created automatically. Which contains by name the name of the document and its pieces, divided by an array of bytes. This function is automated, so it does not require any skills other than how to insert data into the database. For clarity, the essence of the task was shown, in which data will be stored to consolidate the acquired material, or to test their knowledge on a given topic. In non-relational databases, this entity could be implemented within the Lesson entity. Since it is dependent, only on the essence of Lesson, its data will not violate the integrity of this data in any situations.

Tasks can be in two forms: write the result of the work of your program or a text answer to a question. Also, users can recover their passwords using email addresses. In authentication, saving the password in the database will use the hash function md5 or md6. This will help to strengthen the autho-rization and authentication of the user. In the component documents will be stored various types of documents (printed versions or media files). This means that the user can download a convenient version of the lecture or lesson, a document, presentation or video.

On demand, the system should, be quick to process user requests, and return more relevant search results. Also, the system should process requests for downloading and downloading a file in the amount of 20 MB. This space is sufficient for a printed version of the lecture (doc, docx, pdf, etc.) or a presentation of 30 pages with images enclosed in it. This data format is sufficient to explain one topic. Also, the system should have an easy configuration for both the user, and for the teachers and administrator of the performance of their duties. Database queries.

The database will use query-select and request-action. Namely, from the query-sample:

– Output the contents of the table, by criterion or without
– Output of a table with a cross query (Join)
– Query with a calculated field From the query-action

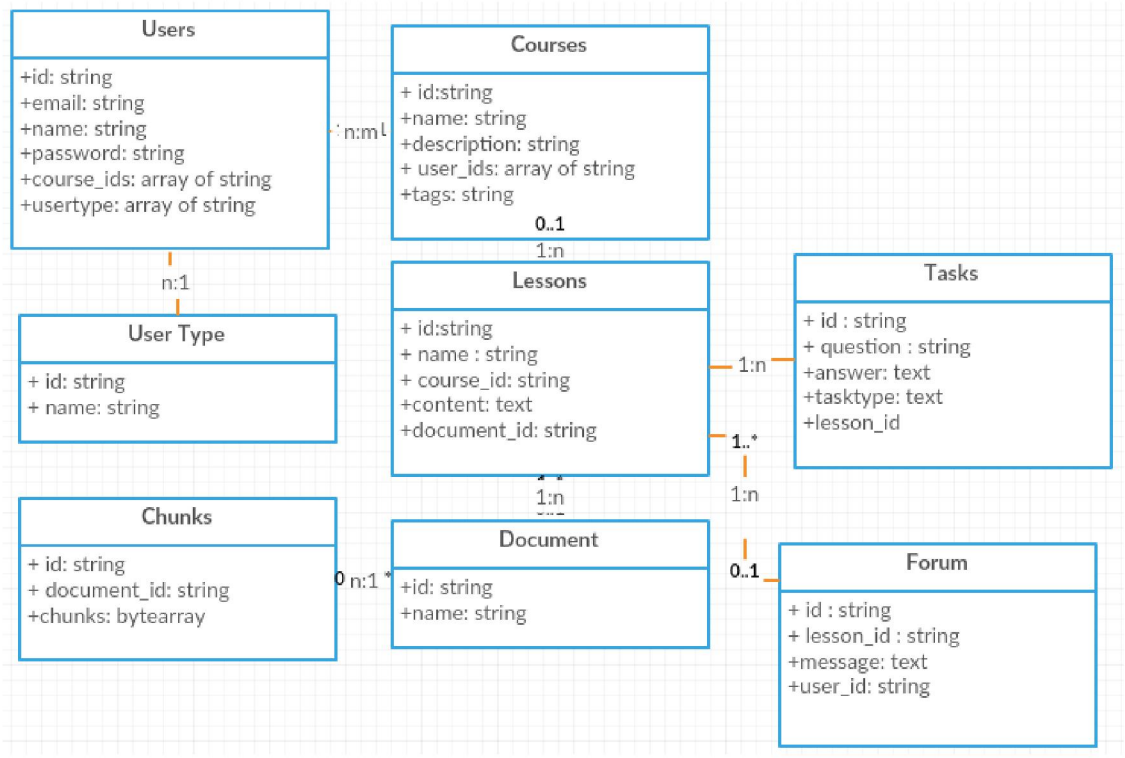

Figure 5 – Database structure

Specific ER diagram for MongoDB DBMS shown in figure 8. There used “many-to-many” relation between tables and deleted necessary entities on class general class diagram.

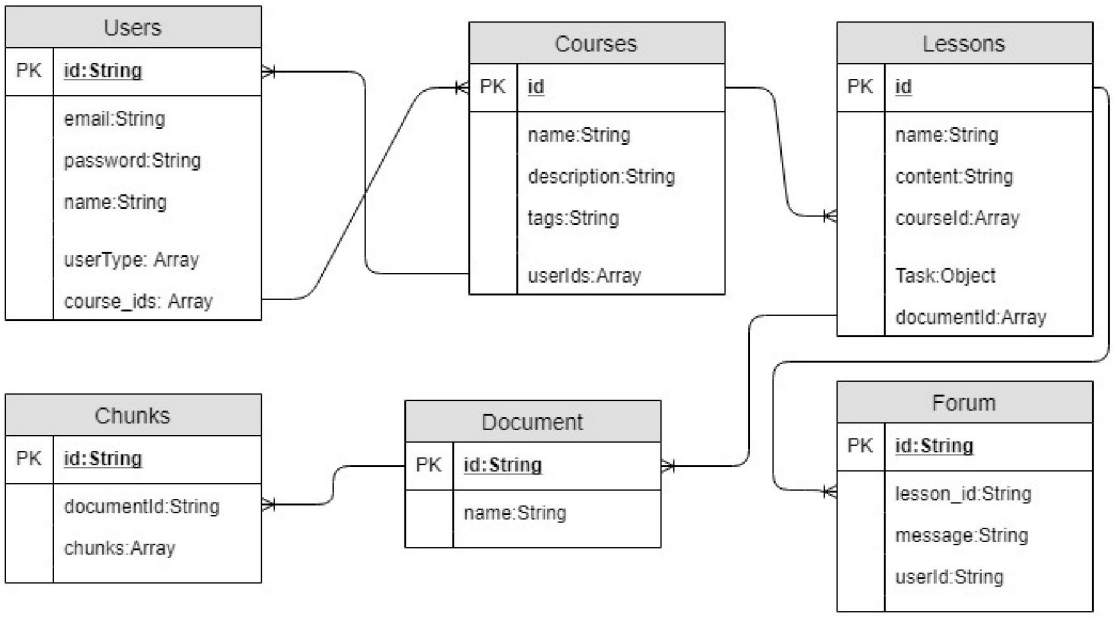

Figure 6 – ER diagram for MongoDB

Realization of online learning system. The system will have several components: the Client part, the Database, as well as the web server or simply the server. Each component will have its own functions. As you can see in the picture above, the client has the following functions: registration - will provide the functionality of entering data into the database as a new user, the input will be done using a hash function, data will be transferred from the client to the server part, the function of sending client data to the server, followed by updating or adding a record. On the server side, there will be functionality to ensure that user

roles are defined, and a connection to the database. The component user manager provides functions for the user to register, log on, or write to the course. The second component of the server side for database management, provides data for manipulating data in the database and communication between the client part.

Database component will have primitive components for manipulating data: Output, input, update, and delete. These functional provide the same functions in the database for manipulation. Between each of the components are connected. The client part is via HTTP protocol, and the server with the database by MongoDB Driver protocol for connection and work with this database. Based on general class diagram is created specific class diagram for C# language (figure 7).

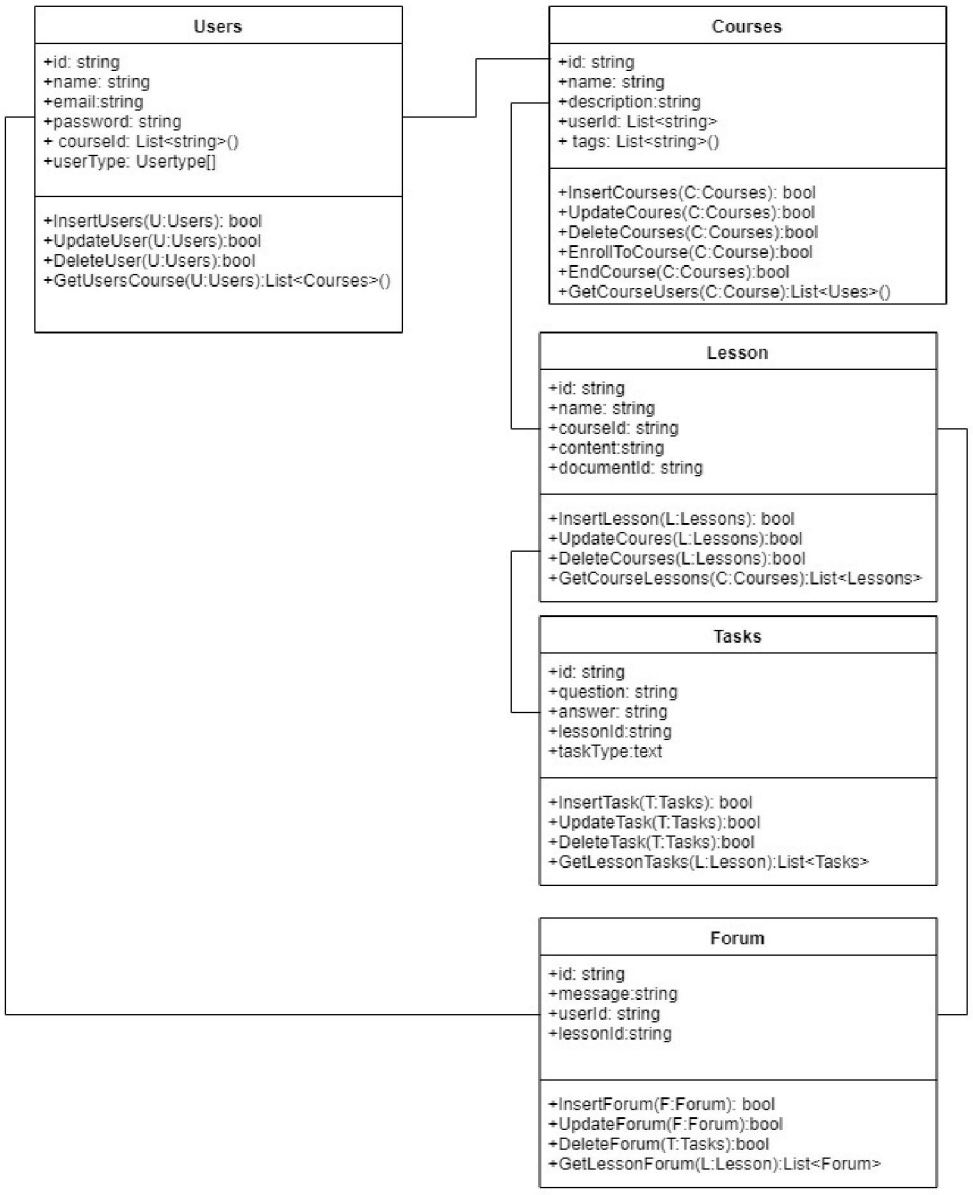

Figure 7 – Class diagram for C# language

The system will use the join operators to cross output the data. From algorithms, the algorithm for sorting an array, built into the programming language functionality. Also, an algorithm for clustering data with archiving. For archiving, the algorithm 7z with the PPMD method will be used, since it showed a more approximate result in the classification of texts, not counting, and the implementation of compression by this algorithm is quite free libraries and easy to implement (SevenZipSharp, Zlib).

Method	R<0.25	R<0.5	R<0.75	R<1.00	R<=1.0
R-index	82.1	86.4	87.1	87.8	89.0
SVM	80.6	83.4	83.5	84.6	85.0
Bzip	56.9	55.2	45.9	51.9	48.2
Gzip	55.7	53.5	53.9	50.1	59.4
Markov chain, order 1	62.3	64.6	63.2	64.3	66.1
Markov chain, order 2	60.9	64.4	61.8	64.7	64.5
Markov chain, order 3	48.6	60.3	59.3	61.7	63.3
RAR	**84.3**	**86.9**	**87.3**	88.5	89.4
PPM, order 2	77.8	79.1	79.4	80.5	81.3
PPM, order 3	80.6	82.3	84.0	85.0	86.4
PPM, order 4	82.5	85.4	86.0	87.7	88.4
PPM, order 5	82.2	86.1	86.3	**88.8**	89.2

Figure 8 – Correlation error of classification by archiving word and le

Analyses of results. As a result, main part of the online learning system was implemented. At the moment, some functionality has been implemented, input and output of files, as well as registration and authorization functional of the system. In the future, it is planned to develop other parts of the system such as classification algorithms, testing trainees, etc. At the moment in the implementation of the system's Web-application used large amount data with following features:

1. Downloading files to the database and retrieving them from the database of files with a size of about 50 MB is processed in a few seconds. This is all thanks to streaming reading and converting the file into byte code on the y. It is more quickly, and we don't do more actions to inserting file into our database than traditional SQL database.

2. Free links between collections (tables). Since it is possible to do both 1-n relations and m-n relations between two collections, without adding any additional collection. This greatly facilitated the connection between the tables.

3. Working with roles is more convenient and at the same time editing in the database is easier than in MSSQL in which all the roles are in a separate table and getting the user role is diffcult.

4. The selection inside the table and the search for text pass very quickly.

5. The integrity of the data is preserved, despite the unstructured data. Depends only on the model that this controller is attached to the base. He also did not concede to the traditional SQL database.

6. MongoDB scalable database. We can use one database scheme for a lot of people. In MSSQL or MySQL we need to reorganize our database scheme or optimize queries. That's why all queries to MongoDB we create on our application.

7. No confusing with JOIN-s. In traditional SQL we confuse with JOIN. Nowadays MongoDB have JOIN to, but many developers not use it, because they can via links and any data from database.

Conclusion. In conclusion, the results of our research suggest that the online learning system can be effectively implemented using the NoSQL database. A comparative analysis of some NoSQL databases, which we conducted and presented above, showed that the choice of MongoDB is preferable, which was due to the simplicity and efficiency of working with this database. In our opinion, after our studies, which are described in this article, it is now simpler and desirable to use an unstructured database for processing large amounts of data. Because in some cases, very strong structures and SQL database designations may not be used. For example, the n-m relationship in SQL requires the inclusion of a new table, so we associate 3 tables. While using NoSQL databases, especially MongoDB, this problem decides to use only two tables with links to each other. In our opinion, this option is more convenient and understandable. Especially, when solving complex problems. It is this feature that will be applied by us in the future to solve complex problems that require processing of large amount unstructured data.

Г. Т. Балакаева[1], Крис Филлипс[2], Д. К. Даркенбаев[1], М. Турдалиев[1]

[1]Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан,
[2]Ньюкасл университеті, Ньюкасл, Ұлыбритания

NoSQL ҚОЛДАНЫП ҮЛКЕН КӨЛЕМДІ ҚҰРЫЛЫМДЫҚ ЕМЕС ДЕРЕКТЕРДІ ӨҢДЕУ

Аннотация. Мақалада үлкен көлемді деректерді өңдейтін заманауи технологияларға талдау жасалды. Авторлар қазіргі таңда сұранысқа ие болып отырған (видео, аудио, анимация және диаграмма т.б. түрде кездесетін файлдар) құрылымды емес үлкен көлемді деректерді өңдеу үшін, NoSQL дерекқорының актуалды

технологияларын қолданды. Авторлар ұсынған NoSQL-дің дерекқорының салыстырмалы талдаулары, MongoDB-дің қарапайым, жұмыс істеуге ыңғайлы дерекқор екендігін және оны таңдау тиімді екендігін көрсетті. Авторлардың пікірінше, мақалада ұсынылған зерттеулерден кейін үлкен көлемді деректерді өңдеуге арналған құрылымды емес дерекқорды пайдалану тиімді әрі қажетті. Мақалада дерекқор интерфейсін әзірлеу, орналастыру сұлбаларын әзірлеу, NoSQL-дегі деректерді тексеру және интеграциялау мен нақты веб-қосымшаларды жасау ұсынылған. NoSQL дерекқорларын, әсіресе MongoDB дерекқорларын пайдаланғанда тек екі кестені бір-біріне сілтеме жасай аламыз. Біздің ойымызша, бұл нұсқа күрделі мәселелерді шешу кезінде ыңғайлы әрі түсінікті. Дәл осы функцияны алдағы уақытта авторлар, үлкен көлемді құрылымдық емес деректерді өңдеуде және көптеген күрделі есептерді шешуде қолданатын болады.

Түйін сөздер: үлкен деректерді өңдеу, құрылымдық емес деректер, NoSQL, веб-қосымша.

Г. Т. Балакаева[1], Крис Филлипс[2], Д. К. Даркенбаев[1], М. Турдалиев[1]

[1]Казахский национальный университет им. аль-Фараби, Алматы, Казахстан,
[2]Университет Нькасла, Ньюкасл, Великобритания

ИСПОЛЬЗОВАНИЕ NoSQL ДЛЯ ОБРАБОТКИ НЕСТРУКТУРИРОВАННЫХ БОЛЬШИХ ДАННЫХ

Аннотация. В статье представлен анализ современных технологий обработки больших данных. Для обработки неструктурированных больших объемов данных, которые сейчас крайне востребованы (данные в виде видео и аудио файлов, анимации, диаграмм и т. д.), авторы использовали актуальные технологии на базе NoSQL. Сравнительный анализ некоторых баз данных NoSQL, которые авторы провели и представили, показал, что выбор MongoDB предпочтительнее, что объясняется простотой и эффективностью работы с этой базой данных. По мнению авторов, после их исследований, которые описаны в этой статье, теперь проще и желательно использовать неструктурированную базу данных для обработки больших объемов данных. В данной статье представлены результаты разработки интерфейсов баз данных, разработки схем развертывания, проверки достоверности и интеграции данных на NoSQL, создания реальных веб-приложений. При использовании баз данных NoSQL, особенно MongoDB, можно использовать только две таблицы со ссылками друг на друга. На наш взгляд, этот вариант более удобен и понятен, особенно при решении сложных задач. Именно эта функция будет применяться авторами в будущем для решения сложных задач, требующих обработки большого количества неструктурированных данных.

Ключевые слова: обработка больших данных, неструктурированные данные, NoSQL, Веб-приложение.

Information about authors:

Balakayeva Gulnar Tultayevna, Doctor of Physical and Mathematical Sciences, Professor of the Computer Science Department of the Al-Farabi Kazakh National University, Faculty of Information Technologies, Almaty, Kazakhstan; gulnardtsa@gmail.com; https://orcid.org/0000-0001-9440-2171

Phillips Christofer, PhD, Professor; University of Newcastle upon Tyne, Newcastle, Great Britain; chris.phillips@newcastle.ac.uk; https://orcid.org/0000-0002-2470-1659

Darkenbayev Dauren Kadyrovich, PhD student, al-Farabi Kazakh National University, Faculty of Information Technologies, Almaty, Kazakhstan; dauren.kadyrovich@gmail.com; https://orcid.org/0000-0002-6491-8043

Turdaliyev Medet, Master degree student, Al-Farabi Kazakh National University, Faculty of Information Technologies, Almaty, Kazakhstan; t_medet@mail.ru; https://orcid.org/0000-0002-2401-0700

REFERENCES

[1] Coursera. Coursera | Online Courses and Credentials From Top Educators. Join for Free. [online], Available at: https://coursera.org/ [Accessed 28 July. 2018].

[2] Moodle.org. (2018). Moodle – Open-source learning platform | Moodle.org. [online], Available at: https://moodle.org/ [Accessed 27 July. 2018].

[3] Stepik A, Free Online Courses. [online] , Available at: https://welcome.stepik.org/ [Accessed 25 July. 2018].

[4] Apache Hadoop. [online], Available at: https:// www. hadoop.apache.org / [Accessed 28 July. 2018].

[5] MongoDB for GIANT Ideas | MongoDB [online], Available at: https:// www. mongodb.com / [Accessed 28 July. 2018].

[6] Apache Cassandra. [online], Available at: https:// www. cassandra.apache.org / [Accessed 28 July. 2018].

[7] Balakayeva G.T., Nurlybayeva K. Simulation of Large Data Processing for Smarter Decision Making // AWER Procedia Information Technology & Computer Science, 3rd World Conference on Information Technology (WCIT-2012). 2013. Vol. (03). P. 1253-1257.

[8] Sagynganova I.K., Markin V.B. // News of the National academy of sciences of the Republic of the Kazakhstan. Series of geology and technical sciences. 2019. Vol. 1, N 433. P. 63-67 (in Eng.). ISSN 2518-170X (online), ISSN2224-5278(Print). https://doi.org/10.32014/2019.2518-170X.7