

NOSQL MA'LUMOTLAR BAZASI VA UNING ASOSIY PRINTSIPLARI

Onarqulov Maqsadjon Karimberdiyevich

*Farg'ona davlat universiteti amaliy matematika va informatika
kafedrası dotsenti*

maxmaqsad@gmail.com

Ismoiljonova Odina Isroiljon qizi

Farg'ona davlat universiteti 3-kurs talabasi

ismoiljonovaodina88@gmail.com

Annotatsiya: Ushbu maqola NoSQL ma'lumotlar bazalarining asosiy tamoyillari va xususiyatlarini o'rganishga bag'ishlangan. NoSQL — katta hajmdagi va tez o'zgaruvchan ma'lumotlarni saqlash, boshqarish va qayta ishlash uchun yaratilgan zamonaviy ma'lumotlar bazasi texnologiyasidir. Maqolada NoSQL tizimlarining asosiy prinsiplari, jumladan, yuqori masshtablash imkoniyati, moslashuvchan struktura, yuqori ishlash tezligi, mavjudlik va ishonchlilik haqida tushuntiriladi. Shuningdek, NoSQL ma'lumotlar bazalarining turlari, ya'ni kalit-qiyamat, hujjatli, graf, va kolonkalarga asoslangan bazalar keltirib o'tilgan. Mazkur yondashuvlar katta hajmdagi va real vaqtda qayta ishlash talab qilinadigan tizimlar uchun keng qo'llanilishi tufayli, NoSQL bazalari zamonaviy dasturiy ta'minot sohasida muhim ahamiyat kasb etadi.

Kalit so'zlar: NoSQL, Ma'lumotlar bazasi, Katta ma'lumotlar, Masshtablash, Hujjatli bazalar, Kalit-qiyamat bazalari (Key-Value Stores), Hujjatli bazalar (Document Stores), Graf bazalari (Graph Databases), Kolonkalarga asoslangan bazalar (Column-Family Stores)

Аннотация: Эта статья посвящена исследованию основных принципов и характеристик NoSQL баз данных. NoSQL — это современная технология баз данных, созданная для хранения, управления и обработки больших объемов и быстро меняющихся данных. В статье объясняются

основные принципы NoSQL систем, включая высокую масштабируемость, гибкую структуру, высокую производительность, доступность и надежность. Также рассматриваются типы NoSQL баз данных, такие как ключ-значение, документо-ориентированные, графовые и колонковые базы данных. Эти подходы, благодаря их широкому применению в системах, требующих обработки больших данных и работы в реальном времени, делают NoSQL базы данных важными в современном программном обеспечении.

Ключевые слова: *NoSQL, База данных, Большие данные, Масштабируемость, Документо-ориентированные базы данных, Базы данных типа ключ-значение (Key-Value Stores), Документо-ориентированные базы данных (Document Stores), Графовые базы данных (Graph Databases), Колонковые базы данных (Column-Family Stores).*

Abstract: *This article is dedicated to exploring the fundamental principles and characteristics of NoSQL databases. NoSQL is a modern database technology designed for storing, managing, and processing large-scale and rapidly changing data. The article explains the core principles of NoSQL systems, including high scalability, flexible structure, high performance, availability, and reliability. Additionally, the types of NoSQL databases, such as key-value, document, graph, and column-family stores, are discussed. These approaches, due to their widespread use in systems that require large-scale and real-time data processing, make NoSQL databases highly significant in modern software development.*

Keywords: *NoSQL, Database, Big Data, Scalability, Document Stores, Key-Value Stores, Document Databases, Graph Databases, Column-Family Stores.*

NoSQL nima va NoSQL ma'lumotlar bazasi nima?

NoSQL ma'lumotlar bazasi texnologiyasi relyatsion ma'lumotlar bazalari tomonidan ishlatiladigan ustunlar va qatorlar o'rniga JSON hujjatlarida

ma'lumotlarni saqlaydi. Aniqroq qilib aytadigan bo'lsak, NoSQL "nafaqat SQL" emas, balki "nafaqat SQL" degan ma'noni anglatadi. Bu shuni anglatadiki, NoSQL JSON ma'lumotlar bazasi tom ma'noda "SQL yo'q" yordamida ma'lumotlarni saqlashi va olishi mumkin. Yoki siz ikkala dunyoning eng yaxshisi uchun JSON moslashuvchanligini SQL kuchi bilan birlashtira olasiz. Shunday qilib, NoSQL ma'lumotlar bazalari moslashuvchan, kengaytiriladigan va zamonaviy korxonalarining ma'lumotlarni boshqarish talablariga tezda javob bera oladigan qilib qurilgan. Quyidagilar NoSQL ma'lumotlar bazasining to'rtta eng mashhur turini belgilaydi:

1. Hujjat ma'lumotlar bazalari asosan ma'lumotlarni hujjatlar sifatida saqlash uchun qurilgan, shu jumladan, lekin ular bilan cheklanmagan JSON hujjatlari. Ushbu tizimlar NoSQL ma'lumotlar bazasi misolida XML hujjatlarini saqlash uchun ham ishlatilishi mumkin.

2. Kalit-qiymati oson qidirish uchun noyob kalitlar bilan identifikatsiyalangan yozuvlar bilan to'plamlarda bog'langan ma'lumotlarni guruhlaydi. Kalit-qiymatli do'konlar NoSQL ma'lumotlar bazasi strukturasining afzalliklarini saqlab qolgan holda, relyatsion ma'lumotlar bazalarining qiymatini aks ettirish uchun etarli tuzilishga ega (norelational ma'lumotlar bazalaridan farqli o'laroq).

3. Keng ustunli ma'lumotlar bazalari relyatsion ma'lumotlar bazalarining jadval formatidan foydalanadi, lekin ma'lumotlarning har bir satrda, hatto bitta jadvalda ham nomlanishi va formatlanishi bo'yicha keng farqlarga imkon beradi. Kalit-qiymatli do'konlar singari, keng ustunli ma'lumotlar bazalari ham juda ko'p moslashuvchanlikni saqlagan holda ba'zi asosiy NoSQL tuzilmalariga ega.

4. Grafik ma'lumotlar bazalari saqlangan ma'lumotlar nuqtalari orasidagi munosabatlarni aniqlash uchun grafik tuzilmalaridan foydalanadi. Grafik ma'lumotlar bazalari tuzilmagan va yarim tizimli ma'lumotlardagi naqshlarni aniqlash uchun foydalidir.

NoSQL ma'lumotlar bazalarining 5 turi

NoSQL ma'lumotlar bazalarining beshta asosiy turi mavjud:

1. Kalit-qiyamat bazalari (Key-Value Stores)

Bu turdagi ma'lumotlar bazalarida ma'lumotlar kalit-qiyamat juftligi sifatida saqlanadi. Redis va Riak kabi tizimlar ushbu turdagi ma'lumotlar bazalariga misol bo'ladi. Kalit-qiyamat bazasi ma'lumotlar bazasining sodda turi bo'lib, unda har bir element kalit va qiymatlarni o'z ichiga oladi. Har bir kalit noyobdir va bitta qiymat bilan bog'liq. Ular keshlash va sessiyalarni boshqarish uchun ishlatiladi va o'qish va yozishda yuqori samaradorlikni ta'minlaydi, chunki ular narsalarni xotirada saqlashga moyildirlar.

2. Hujjatli bazalar (Document Stores)

Hujjatga yo'naltirilgan ma'lumotlar bazasi JSON (JavaScript Object Notation), BSON yoki XML ob'ektlariga o'xshash hujjatlardagi ma'lumotlarni saqlaydi. MongoDB va CouchDB hujjatli bazalarga misol bo'la oladi. Bu usulda har bir hujjat turli xildagi ma'lumotlar to'plamini o'zida mujassam etishi mumkin. Har bir hujjatda juft maydonlar va qiymatlar mavjud. Qiymatlar odatda turli xil bo'lishi mumkin, jumladan qatorlar, raqamlar, mantiqiy qiymatlar, massivlar yoki hatto boshqa ob'ektlar. Hujjatlar ma'lumotlar bazasi moslashuvchan ma'lumotlar modelini taklif qiladi, bu yarim tuzilgan va odatda tuzilmagan ma'lumotlar to'plamlari uchun juda mos keladi. Ular, shuningdek, ichki o'rnatilgan tuzilmalarni qo'llab-quvvatlaydi, bu murakkab munosabatlar yoki ierarxik ma'lumotlarni taqdim etishni osonlashtiradi.

3. Graf bazalari (Graph Databases)

Graf ma'lumotlar bazalari graf tuzilmalarini saqlash uchun ishlatiladi. Masalan, ijtimoiy tarmoqdagi odamlar orasidagi aloqalar graf orqali ifodalanishi mumkin. Neo4j va ArangoDB graf bazalariga misol bo'ladi. Grafik ma'lumotlar bazasi ma'lumotlarni tugunlar va qirralar ko'rinishida saqlaydi. Tugunlar odatda odamlar, joylar va narsalar (masalan, otlar) haqidagi ma'lumotlarni saqlaydi, qirralar esa tugunlar orasidagi munosabatlar haqida ma'lumotni saqlaydi. Ular yuqori darajada bog'langan ma'lumotlar uchun yaxshi ishlaydi, bu erda

munosabatlar yoki naqshlar dastlab unchalik aniq bo'lmasligi mumkin. Grafik ma'lumotlar bazalariga misollar Neo4J va Amazon Neptune. MongoDB hamgrafiklarni o'tish imkoniyatlarini beradiagregatsiya quvurining \$graphLookup bosqichidan foydalanish.

4. Kolonkalariga asoslangan bazalar (Column-Family Stores)

Bu turdagi bazalar ma'lumotlarni kolonkalariga ajratib saqlaydi. Apache Cassandra va HBase kabi ma'lumotlar bazalari kolonka-oilaviy (column-family) bazalar sifatida ishlatiladi. Ko'p modeli ma'lumotlar bazalari NoSQL ma'lumotlar modellarining bir nechta turlarini qo'llab-quvvatlaydi, shuning uchun ishlab chiquvchilar o'zlarining ilovalari talablari asosida tanlashlari mumkin. Ushbu ma'lumotlar bazalari ma'lumotlar bazasi misolida bir nechta ma'lumotlar modellarini boshqara oladigan yagona ma'lumotlar bazasi mexanizmiga ega. Misollar CosmosDB va ArangoDB.

NoSQL ma'lumotlar bazalarining asosiy prinsiplari

1. Masshtablashning yuqori darajasi (Scalability)

NoSQL tizimlari keng masshtablash imkoniyatiga ega bo'lib, katta hajmdagi ma'lumotlarni tezkor ishlash imkoniyatini ta'minlaydi. NoSQL tizimlari gorizontallik masshtablashni qo'llab-quvvatlaydi, ya'ni ma'lumotlar bazasini kengaytirish uchun serverlar qo'shilishi mumkin. Bu tizimning resurslarini kengaytirish imkoniyatini anglatadi, ya'ni u ma'lumotlar hajmi, foydalanuvchilar soni yoki boshqa yuklamalar oshganda, o'z ish faoliyatini saqlab qolishi yoki yanada samaraliroq bo'lishi mumkin. Bu printsip, ayniqsa, katta hajmdagi ma'lumotlar bilan ishlovchi tizimlar, jumladan, NoSQL ma'lumotlar bazalari uchun juda muhimdir.

Masshtablashning turlari

1. Gorizontallik masshtablash (Horizontal Scalability)

Ta'rif: Gorizontallik masshtablash, ma'lumotlar bazasiga yangi serverlar (nodlar) qo'shish orqali tizimni kengaytirishdir. Bu yondashuv, ko'proq resurslarni taqdim etish va yuklamani bir necha serverga taqsimlash orqali ishlash tezligini oshirish imkonini beradi.

Afzalliklari:

1. Tez kengayish imkoniyati.
2. Har bir yangi nodni qo'shish oson va arzon bo'lishi mumkin.
3. Tizim ishdan chiqqan taqdirda, boshqa serverlar orqali davomiy ishlashni ta'minlaydi.

2. Vertikal masshtablash (Vertical Scalability)

Ta'rifi: Vertikal masshtablash, mavjud serverning resurslarini (masalan, protsessor, xotira, saqlash joyi) oshirish orqali kengaytirishni anglatadi. Bu usul, odatda, serverning maksimal imkoniyatlariga yetguncha ishlatiladi.

Afzalliklari:

1. Oson implementatsiya, chunki dasturiy ta'minot ko'pincha o'zgarmaydi.
2. Tizimning samaradorligini oshirish imkonini beradi.

Kamchiliklari:

1. Har qanday serverning resurslari chegaralangan.
2. Odatda, yangi apparat xarajatlari yuqori bo'lishi mumkin.

NoSQL ma'lumotlar bazalarida masshtablash

NoSQL ma'lumotlar bazalari gorizontal masshtablashni qo'llab-quvvatlaydi, bu esa ularga quyidagi afzalliklarni beradi:

Katta hajmdagi ma'lumotlar bilan ishlash: NoSQL tizimlari katta hajmdagi tuzilgan va tuzilmagan ma'lumotlarni tezda qayta ishlash qobiliyatiga ega.

Tez o'zgaruvchan muhitga moslashuvchanlik: Foydalanuvchilar soni yoki ma'lumotlar hajmi oshganda tizimni kengaytirish oson.

Yuqori mavjudlik: Bir nechta serverlar o'rtasida ma'lumotlar tarqatilganligi sababli, bitta server ishdan chiqsa, tizim faoliyatini davom ettiradi.

Umuman olganda, masshtablashning yuqori darajasi NoSQL ma'lumotlar bazalarini an'anaviy SQL tizimlariga nisbatan o'ziga xos va samarali yechimlar sifatida ajratib turadi, ayniqsa, katta ma'lumotlar bilan ishlovchi zamonaviy dasturiy ta'minotlar uchun.

2. Moslashuvchan ma'lumotlar tuzilmasi (Flexible Schema)

NoSQL bazalarida oldindan belgilangan qat'iy ma'lumotlar strukturasi ega bo'lmaganligi sababli, ma'lumotlar turli formatlarda saqlanishi mumkin. Bu esa dasturchilarga ma'lumotlar tuzilmasini tezda o'zgartirish imkonini beradi, ayniqsa, ma'lumotlar tez-tez o'zgarib tursa. Moslashuvchan ma'lumotlar tuzilmasi (Dynamic Data Structure) — bu ma'lumotlarni saqlash va ular bilan ishlash uchun moslashuvchan, ya'ni dinamik ravishda o'zgartiriladigan tuzilmalardir. Bu tuzilmalar dastur ishga tushganidan keyin o'zgarishi mumkin, ya'ni ularni yaratish, o'chirish yoki o'zgartirishni dastur davomida amalga oshirish mumkin.

Misollar:

1. Bog'langan ro'yxatlar (Linked List): Bu tuzilma, har bir element (noma'lum uzunlikdagi ro'yxatdagi node) o'zidan keyingi elementga (node) havola saqlaydi. Bog'langan ro'yxatlar ro'yxatni kengaytirish yoki qisqartirish uchun yaxshi moslashuvchanlikka ega.

2. Stak (Stack) va Queue (Queue): Bu ma'lumotlar tuzilmalari faqat bir tomonlama ma'lumot qo'shish va o'chirish imkonini beradi. Ular odatda dinamik tarzda ishlaydi, ya'ni elementlar soni dastur davomida o'zgarishi mumkin.

3. Dinamik massivlar (Dynamic Arrays): Bularga misol sifatida, masalan, C++ yoki Java dasturlash tillarida mavjud bo'lgan Vector (C++) yoki ArrayList (Java) ma'lumotlar tuzilmalarini keltirish mumkin. Dinamik massivlar, belgilangan o'lchamga ega bo'lishi mumkin, ammo ularning o'lchami dastur ishlash jarayonida o'zgartirilishi mumkin.

4. Xesh-jadval (Hash Table): Xesh-jadvalda ma'lumotlar kutilgan kalitlar yordamida indekslanadi, va bu tuzilmada ma'lumotlar dinamik tarzda qo'shilishi yoki o'chirilishi mumkin.

5. Darbni jadvallari (Trees) va grafiklar (Graphs): Ular ko'pincha dinamik ravishda o'sishi mumkin, ya'ni yangi tugunlar qo'shish yoki mavjud tugunlarni o'chirish mumkin.

Dinamik ma'lumotlar tuzilmalarining asosiy afzalligi shundaki, ular ma'lumotlar soni oldindan ma'lum bo'lmagan hollarda samarali ishlaydi va tizim resurslarini yanada optimallashtirishga yordam beradi.

3. Ma'lumotlarning yuqori ishlash tezligi (High Performance)

Ma'lumotlarni yuqori ishlash tezligida saqlash va ishlov berish, ma'lumotlar strukturasi optimalligi, algoritmlarning samaradorligi va tizim resurslarining to'g'ri taqsimlanishiga bog'liq. Yuqori ishlash tezligini ta'minlash uchun quyidagi usullarni va texnikalarni ko'rib chiqish mumkin:

1. Optimal Ma'lumotlar Tuzilmasini Tanlash

Ma'lumotlar tuzilmasi tanlovi tezlikka katta ta'sir ko'rsatadi. Har bir tuzilma o'zining samarali ishlash vaqtiga ega bo'lgan operatsiyalarni taqdim etadi. Masalan:

Hash Table (Xesh-jadval): Odatda $O(1)$ vaqtda elementga kirish, qo'shish yoki o'chirish mumkin, bu juda tezkor operatsiyalarni ta'minlaydi.

Binary Search Tree (Ikki tarmoqli qidiruv daraxti): $O(\log n)$ vaqt ichida ma'lumotlarni qidirish yoki qo'shish mumkin.

B-trees: Katta hajmdagi ma'lumotlar bazalarida qo'llaniladi va diskga kirish tezligini optimallashtiradi, bu esa katta ma'lumotlar bilan ishlashda juda samarali.

2. Ma'lumotlarni Keshda Saqlash

Ma'lumotlarni keshda saqlash orqali tezkor kirish imkoniyatini yaratish mumkin. Kesh (cache) tez-tez ishlatiladigan ma'lumotlarni saqlaydi, bu esa tizimning umumiy ish faoliyatini yaxshilaydi. Misol uchun:

In-memory cache: Redis yoki Memcached kabi in-memory kesh tizimlari tezkor ma'lumotlar saqlash va qayta ishlashda ishlatiladi.

CPU Cache: Tezkor xotira (L1, L2, L3) protsessorda saqlanadigan eng tezkor ma'lumotlar.

3. Parallel va Pishirish Texnikalari

Parallel Processing (Paralel ishlov berish): Katta hajmdagi ma'lumotlarni tahlil qilishda bir nechta protsessor yadro va/yoki serverlar yordamida ma'lumotlarni bir vaqtning o'zida qayta ishlash mumkin.

Multithreading: Ma'lumotlarni parallel ravishda qayta ishlash uchun ko'p iplar yordamida bir nechta vazifalarni bir vaqtda bajarish.

4. Effektiv Algoritmalar

Yuqori tezlikdagi ishlov berish uchun samarali algoritmlarni qo'llash kerak. Masalan:

Sortlash algoritmlari: Ma'lumotlarni tez va samarali tartiblash uchun QuickSort yoki MergeSort kabi algoritmlar ishlatiladi, ular $O(n \log n)$ samaradorlikka ega.

Qidiruv algoritmlari: Binom qidiruv (Binary Search) algoritmi, agar ma'lumotlar tartiblangan bo'lsa, $O(\log n)$ vaqt ichida ma'lumotlarni topishga imkon beradi.

5. Disk Operatsiyalarini Minimallashtirish

Disk operatsiyalari juda sekin bo'lishi mumkin, shuning uchun ma'lumotlarni tezkor xotirada saqlash va diskka kamroq murojaat qilish maqsadga muvofiq. Buning uchun:

SSD (Solid State Drive): HDD (Hard Disk Drive) ga qaraganda SSD-lar tezroq ishlaydi va ma'lumotlarga tezroq kirish imkonini beradi.

Prefetching va Buffering: Ma'lumotlar diskdan oldindan yuklab olinadi yoki keshda saqlanadi, bu esa kirishni tezlashtiradi.

6. Indeksatsiya

Ma'lumotlar bazasida indekslar yaratish orqali qidiruv tezligini sezilarli darajada oshirish mumkin. Masalan:

B-Tree yoki Hash Index: Ma'lumotlar bazasidagi indekslar orqali tezkor qidiruv va yangilanishlar amalga oshiriladi.

7. Lazy Loading va Caching

Lazy Loading: Kerak bo'lganda faqat zarur ma'lumotlarni yuklash usuli. Bu tizimning ishlashini sezilarli darajada tezlashtiradi, ayniqsa katta hajmdagi ma'lumotlar bilan ishlashda.

Data Caching: Avvaldan hisoblangan yoki qidirilgan ma'lumotlarni keshga saqlash orqali ularni tezroq olish.

8. Veri Streaming (Ma'lumot oqimi)

Agar ma'lumotlar juda katta hajmda bo'lsa, ularni to'liq yuklab olish o'rniga, oqim (streaming) texnologiyalari orqali tezda ishlov berish mumkin. Bu xususiyat katta video yoki ma'lumotlar to'plamlarini real vaqt rejimida qayta ishlashda ishlatiladi.

4. Yuqori mavjudlik va ishonchlilik (High Availability and Reliability)

Yuqori mavjudlik (High Availability, HA) va ishonchlilik (Reliability) — tizimlarning ishlashini uzluksiz ta'minlash va xatoliklar yoki nosozliklar holatlarida ularning ishdan chiqmasligi uchun mo'ljallangan tushunchalar. Bu tushunchalar katta hajmdagi ma'lumotlarni qayta ishlash, xizmatlarni taqdim etish va tizim resurslarini samarali boshqarishda muhim ahamiyatga ega.

1. Yuqori Mavjudlik (High Availability)

Yuqori mavjudlik, tizim yoki xizmatning uzoq vaqt davomida uzluksiz ishlashini ta'minlash uchun yaratilgan muhitni anglatadi. Bu, tizimning nosozliklar yoki uzilishlarga qarshi bardoshlilikini va zaxiraviy tizimlar orqali xizmatning davomiyligini ta'minlashni o'z ichiga oladi.

2. Ishonchlilik (Reliability)

Ishonchlilik tizimning ishdan chiqmasdan uzoq vaqt davomida xizmat ko'rsatishini ta'minlashga qaratilgan. Tizim ishonchli bo'lishi uchun uning xatolarga qarshi bardoshlilik, uzilishlarni tezda bartaraf etish va ishlashda o'zgarishlarga moslashuvchan bo'lishi muhim.

3. Yuqori Mavjudlik va Ishonchlilikni Ta'minlash uchun Asosiy Texnikalar

Virtualizatsiya: Bir nechta virtual mashinalar yordamida tizimni mustahkamlash va zaxira tizimlar yaratish.

Containerization: Docker kabi texnologiyalar yordamida tizimni izolyatsiya qilish va har bir xizmatni alohida konteynerda ishlatish.

Microservices Architecture: Har bir xizmatni alohida va mustaqil qilib tashkil etish, bu esa biror xizmat ishdan chiqqanda boshqa xizmatlar ishlashda davom etishini ta'minlaydi.

Xulosa:

NoSQL ma'lumotlar bazalari katta hajmdagi va o'zgaruvchan ma'lumotlarni boshqarish uchun an'anaviy SQL tizimlariga muqobil yechim hisoblanadi. Bu tizimlarning asosiy ustunliklari yuqori masshtablash, tezkor ishlash, moslashuvchan tuzilma, yuqori mavjudlik va ishonchlikdir. NoSQL bazalari o'zida kalit-qiyamat, hujjatli, graf va kolonkalarga asoslangan ma'lumotlar bazalarini jamlaydi, bu esa ularni keng ko'lamdagi dasturiy yechimlarda, jumladan, katta hajmdagi ma'lumotlarni real vaqtda qayta ishlash talab etiladigan tizimlarda qo'llash imkonini beradi. Ushbu maqolada NoSQL bazalarining turlari va ularning asosiy tamoyillari yoritilib, zamonaviy texnologik rivojlanish jarayonida bu tizimlarning ahamiyati ko'rsatib o'tilgan.

FOYDALANILGAN ADABIYOTLAR:

Kitoblar

1. Stonebraker, M., & Cattell, R. (2010). "SQL vs. NoSQL: SQL is not the new NoSQL." Communications of the ACM.
2. Moniruzzaman, A. B. M., & Hossain, S. A. (2013). "NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison." International Journal of Database Theory and Application.
3. Strauch, C. (2011). "NoSQL Databases." Lecture Notes, Stuttgart Media University.
4. Онаркулов, М. К. (2024). ИНТЕГРАЦИЯ НЕЙРОННЫХ СЕТЕЙ В РАСПОЗНАВАНИИ РЕЧИ И РАЗРАБОТКА НОВЫХ МЕТОДОВ. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 47(7), 193-197
4. Онаркулов, М. К., угли Юсупов, М. А., & угли Умиржонов, Л. А. (2023). ПРИМИНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В РАСПОЗНАВАНИИ РЕЧИ. Educational Research in Universal Sciences, 2(3), 1206-1210.

5. Umarjon o'g, Y. Y. L. (2024). PROBLEMS AND MODERN TRENDS IN COMPUTING ENGINEERING. Multidisciplinary and Multidimensional Journal, 3(6), 17-21.
6. Onarkulov, M., & Nabijonov, A. (2024). DB2: KATTA HAJMDAGI MA'LUMOTLARNI SAMARALI BOSHQARISH. Инновационные исследования в науке, 3(5), 99-104.
7. Onarkulov, M., & G'oyibova, G. (2024). SQL SERVER ILOVASIDA UNIVERSAL MODELNI QO'LLASH VA KONVERTATSIYA ETISHNING SAMARALI YO'LLARI. Академические исследования в современной науке, 3(18), 147-152.
8. Karimberdiyevich, M. O. (2024). RBF TURLARINING O 'QITISH ALGORITMI VA XOR MASALASI. worldly knowledge conferens, 8(1), 176-180.
9. Karimberdiyevich, M. O. (2024). GAUSS FUNKSIYASI. worldly knowledge conferens, 8(1), 239-244.
10. Karimberdiyevich, O. M. (2024). SQL TILIDA PROTSEDURA VA FUNKSIYALARDAN FOYDALANISHNING AHAMIYATI. worldly knowledge conferens, 8(1), 145-148.
11. Karimberdiyevich, O. M. (2024). BIR QATLAMLI NEYRON TO'RLARI VA ULARNI YARATISH USULLARI. IQRO INDEXING, 9(2), 104-108.

Veb-saytlar

1. MongoDB: <https://www.mongodb.com>
2. Cassandra: <https://cassandra.apache.org>
3. Neo4j: <https://neo4j.com>
4. Couchbase: <https://www.couchbase.com>