

## **СФЕРИ ЗАСТОСУВАННЯ НЕРЕЛЯЦІЙНИХ БАЗ ДАНИХ**

Вінницький національний технічний університет

### **Анотація:**

*Розглянуто основні типи нереляційних баз даних та конкретні приклади їх використання провідними компаніями.*

**Ключові слова:** нереляційний, NoSQL, база даних (БД), Neo4j.

### **Abstract:**

*The main types of non-relational databases and specific examples of their use by leading companies are discussed.*

**Keywords:** non-relational, NoSQL, database (DB), Neo4j.

### **Вступ**

З розвитком ІТ індустрії зростає потреба в нових підходах до вирішення проблеми зберігання та управління великим об'ємом даних. Сьогодні найбільш популярними базами даних є реляційні SQL бази даних, які з'явилися ще у 70-х роках минулого століття. Однак зі стрімким зростанням обсягів зберігання даних, які є погано структурованими, висуваються підвищені вимоги до швидкості пошуку та продуктивності обробки запитів до бази даних. Якщо цілісність даних не відіграє великого значення, а першочерговими характеристиками є гнучкість та швидкодія, то доречніше використовувати нереляційні (NoSQL) бази даних.

NoSQL іноді поділяють на дореляційні (тут NoSQL скоріш за все означає 'не SQL'), тобто ті, які існували до створення мови SQL, та ті, які виникли після неї ('not only SQL' – не тільки SQL). У цій статті ми розглядатимемо останні.

NoSQL ('not only SQL' – не тільки SQL) — база даних, яка забезпечує механізм зберігання та видобування даних відмінний від підходу таблиць-відношень в реляційних базах даних. Подібні бази даних існували вже в другій половині 1960-х років, але тоді вони ще не здобули гучне ім'я "NoSQL", одержане після сплеску популярності на початку 21-го століття, що був спричинений потребами Web 2.0 компаній, такими як Facebook, Google, та Amazon.com. NoSQL бази даних все більше і більше використовуються в задачах із застосуванням великих даних та web-застосунках реального часу. NoSQL системи називають "Not only SQL" для підкреслення того, що вони можуть підтримувати SQL-подібну структуру та мову запитів [1].

За рейтингом баз даних <https://db-engines.com/en/ranking> станом на лютий 2020 в десятку входять тільки три NoSQL бази даних – [MongoDB](#), [Elasticsearch](#) та [Redis](#). Перші місця займають реляційні бази даних.

Незважаючи на простоту та ефективність використання нереляційних баз даних, вони, зазвичай, використовуються, як додаток до реляційних баз даних, зокрема для вирішення окремих проблем, які не раціонально вирішувати засобами SQL [2].

### **Різновиди NoSQL баз даних**

Виділяють чотири основні типи NoSQL баз даних [3]. Вони розрізняються моделлю даних, підходом до розподіленості та реплікації, завдяки чому можуть в різній мірі підходити під ті чи інші види конкретних завдань.

**Бази даних "ключ-значення"** – це найпростіший вид баз даних, який по суті є асоціативним масивом, в якому кожному значенню відповідає свій ключ. Завдяки простоті таких баз даних вони є дуже ефективними при масштабуванні. Оскільки, у базах даних такого виду відсутні зв'язки між значеннями, то кількість оброблюваних даних визначається лише робочими потужностями. Можна

зробити висновок, що такі технології зазвичай цікаві компаніям, які зберігають великі об'єми інформації, якими зазвичай є хмарні сховища.

Подібні бази даних використовуються, коли повністю відсутні зв'язки між окремими осередками сховища. Такі бази даних не повністю підходять для повної заміни реляційних БД, але знайшли своє призначення, наприклад, в кешуванні.

Найбільш відомими прикладами СКБД даного типу є Amazon DynamoDB, Berkeley DB, MemcacheDB, Redis і Riak.

**Документоорієнтована база даних** – це система зберігання ієрархічних структур даних, що має структуру дерева або лісу. Документоорієнтовані БД є більш складною версією баз даних типу “ключ-значення”, але вони ще досі не підходять для систем, у яких є безліч зв'язків між елементами, але дозволяють отримати відповідь на запит, не повністю навантажуючи оперативну пам'ять. Деревоподібна структура дозволяє організовувати колекції документів за типом або тематикою, а за допомогою пошуку можна знаходити як повністю весь документ, так і його частини.

Документоорієнтовані БД знаходять своє призначення у завданнях, де необхідне впорядковане зберігання інформації, але немає зв'язків між ними та не потребують постійного збору статистики.

Приклади СКБД цього типу: CouchDB, Couchbase, MarkLogic, MongoDB, eXist.

**Графова модель баз даних** – це узагальнення мережевої моделі даних і відрізняється лише сильними зв'язками між вузлами. Ідеально підходять для соціальних мереж, так як передбачають природну графову структуру даних. Для великих графів використовують алгоритми, що частково переміщують граф в оперативну пам'ять.

Графові СУБД: ArangoDB, FlockDB, Giraph, HyperGraphDB, Neo4j, OrientDB

**Bigtable-подібні бази даних** містять впорядковані дані у вигляді розрідженої матриці, рядки і стовпці якої використовують як ключі. Ці сховища мають багато спільного з документоорієнтованими БД – системами керування вмістом, агрегаторами реєстрації на події, блоги, розсилки. Bigtable-подібні БД не потрібно плутати з лінійними сховищами, так як лінійні сховища – це по суті реляційні БД з роздільним зберіганням колонок. Bigtable-подібні БД використовують для веб-індексування та подібних задач, які мають величезні обсяги даних.

Відомими прикладами таких баз даних є: HBase, Cassandra, Hypertable, SimpleDB

### **Сфери застосування NoSQL баз даних**

NoSQL бази даних максимально розкривають свій потенціал у розподілених системах. Вагомими перевагами при роботі з розподіленими системами мають всі типи нереляційних сховищ, за винятком графових БД — вони за визначенням передбачають велику кількість зв'язків між вузлами даних [3].

Помітною різницею між SQL та NoSQL є те, що NoSQL бази даних часто фокусуються на наданні кращої доступності операцій завдяки слабшим моделям, з чітким фокусом на стійкість до поломок [4]. Розробникам завжди доводиться йти на компроміс – вибирати між вузькоспеціалізованою NoSQL та широкоспеціалізованою SQL. 75.6% опитаних розробників комбінують SQL та NoSQL [5].

У якості допоміжних NoSQL бази даних є надзвичайно швидкими для виконання одного типу задач (наприклад, робота з графом) та абсолютно непідходящими для вирішення інших (соціальної мережі не потрібна чітка узгодженість, в той час як для банківського додатку вона є невід'ємною).

Так, у системі WindyGrid [6] для моніторингу міста Чикаго використовується нереляційна база даних MongoDB. В ній зберігають великий масив неструктурованих даних: платформа надає сенс мільйонам записів даних, що збираються щодня з 15 найважливіших відділів Чикаго, включаючи поліцію, транспорт та пожежну службу, дорожні роботи, затримки вивезення сміття, інформація про дзвінки до 911 (надзвичайні ситуації) та 311 (скарги про шум), публічні твіти про деталі роботи міста, розташування автобусів на їхньому маршруті, кольори світлофорів в реальному часі та багато іншого. MongoDB також використовується в хмарі Google Cloud як рішення для розробників для швидкого розгортання проектів.

PayPal, світовий лідер у галузі онлайн-платежів, спершу використовував Couchbase (база даних NoSQL) для своєї медійної рекламної мережі Media Network та для побудови міжканальної аналітики користувачів для вирішення задачі профілювання, сегментації, підтвердження справжності особи та ін. До 2014 року компанія керувала понад 1 мільярдом документів та 10-ма терабайтами даних за допомогою Couchbase. PayPal розширив використання Couchbase в аналітиці для отримання інформації про користувачів, обробляючи мільйони оновлень на хвилину завдяки технологіям Kafka та Hadoop.

Couchbase поверх багаторівневої AWS (Amazon Web Services) архітектури використовується VoIP додатком Viber для оновлень профілів користувачів в реальному часі. Також Couchbase використовується низкою відомих брендів: eBay (лістинги товарів в реальному часі), LinkedIn (кешування інформації), Tesco (каталог та менеджмент інвентарю), Cisco (VSRM – технологія для шифрування відеопотоку (для платних телеканалів)) [7].

Низка відомих сервісів використовують Redis: Github – в якості бази даних “ключ-значення” для маршрутизації інформації [8] (під час HTTP запитів на отримання інформації із серверів); Twitter – для кешування твітів. StackOverflow використовує Redis у якості окремого рівня кешування мережі [9].

NoSQL є також популярним у сфері «інтернету речей» (Internet of things, IoT). За рахунок великої кількості потоків даних (з кожного датчика кожного пристрою), практично повсюдній присутності часових міток, за якими ведуться запити, та нехтування повнотою даних (зникнення сигналу на деякий час) у IoT нереляційні бази даних є привабливими для розробників. Якщо обмін «нормальними» даними проводиться через TCP (мережевий протокол транспортного рівня в основі якого лежить постійний зворотній зв'язок), то дані IoT передаються через UDP [10] (мережевий протокол транспортного рівня, який працює без встановлення зворотного зв'язку “спра-лячи” пакетами даних).

Не всі IoT використовують NoSQL. На думку Метта Бутчера (Matt Butcher), Revolv розробника, за своєю природою дані є відносними, тобто реляційними [11]. Revolv – платформа смарт-будинку, придбана компанією Nest (американським лідером з автоматизації домівок) – перейшла з MongoDB на DynamoDB і згодом на реляційну PostgreSQL.

Neo4j є прикладом графової бази даних. Щоб отримати дані з такої бази даних необхідна мова інша, ніж SQL [12], яка б дозволила легко оперувати графом. Тому за визначенням графові БД є нереляційними та використовують NoSQL.

eBay, наприклад, використовує її для надання покупцям рекомендацій. Наприклад, більшість пошукових систем у запиті “Ми з дружиною ідемо на кемпінг на Синебір наступного тижня, нам потрібен намет” відреагували б на слово “намет”, а додаткова контекстна інформація, що стосується місця, температури, розміру намету і т.д. зазвичай втрачається. eBay це вирішив з допомогою створення власного додатку з використанням Neo4j. Запит постійно еволюціонує – з кожним уточненням параметрів товару Neo4j знаходить найоптимальніші варіанти.

NASA використовує Neo4j для опрацювання “вивчених уроків” (даних із попередніх місій) та для так званих Великих Даних (Big Data). За словами головного архітектора NASA, завдяки отриманим даним було зекономлено 2 роки роботи та близько мільйона доларів податків [13]. Щодо Big Data, то NASA збирає як структуровані, так і неструктуровані дані. Neo4j було застосовано для дослідження взаємозв'язку між відгуками астронавтів після повернення з космічної станції. Це дозволило NASA опрацювати масив даних за останні 15 років та зробити висновки щодо того, на які системи та підсистеми вплинули експерименти, проведені на борту.

Monsanto – агрокультурна компанія, що займається розробкою ГМО (придбана компанією Bayer) використовує Neo4j для обробки інформації про геномні послідовності рослин, щоб покращити селекцію [14]. Для вирішення генетичних проблем дослідники ставляться до даних як до сімейного дерева, в якому набагато більше предків. Через відповідність даних графовій моделі аналіз займає лічені секунди.

Neo4j використовують також такі бренди, як Caterpillar, Airbnb, Lockheed Martin, Walmart, Comcast.

### Висновки

В результаті проведеного аналізу, було виявлено, що NoSQL (нереляційні) бази даних є потужним інструментом обробки великої кількості слабоструктурованих або неструктурованих даних, для яких на першому місці стоїть швидкість обробки, а не повнота даних. Вибір цього інструменту залежить від конкретної задачі. Попри свою зростаючу популярність, NoSQL бази даних все ще є нішевыми, та не можуть повноцінно замінити лідерів галузі – SQL БД. Через відмінність сфер застосування SQL та NoSQL бази даних існуватимуть поряд, доповнюючи одна одну.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. NoSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/NoSQL>

2. Як краще використовувати NoSQL бази даних [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/cloud/blog/how-to-choose-a-database-on-ibm-cloud>
3. Переваги та недоліки нереляційних баз даних [Електронний ресурс] – Режим доступу до ресурсу: <https://www.quality-assurance-group.com/nosql-perevagy-ta-nedoliky-nerelyatsijnyh-baz-danyh/>
4. Слабка ізоляція, розподіл та NoSQL [Електронний ресурс] – Режим доступу до ресурсу: <http://www.redbook.io/ch6-isolation.html>
5. Тренди баз даних 2019 [Електронний ресурс] – Режим доступу до ресурсу: <https://bit.ly/32J11ox>
6. Як Чикаго використовує MongoDB щоб створити розумніше місто [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/customers/city-of-chicago>
7. Клієнти Couchbase [Електронний ресурс] – Режим доступу до ресурсу: <https://www.couchbase.com/customers>
8. Як ми зробили Github швидшим [Електронний ресурс] – Режим доступу до ресурсу: <https://github.blog/2009-10-20-how-we-made-github-fast>
9. Як StackOverflow кешує дані [Електронний ресурс] – Режим доступу до ресурсу: <https://bit.ly/3akc1sI>
10. Яка NoSQL база даних краще зберігає IoT дані [Електронний ресурс] – Режим доступу до ресурсу: <https://bit.ly/2IagYZ2>
11. Більше не NoSQL [Електронний ресурс] – Режим доступу до ресурсу: <http://technosophos.com/2014/04/11/nosql-no-more.html>
12. Використання Neo4j eBay [Електронний ресурс] – Режим доступу до ресурсу: <https://neo4j.com/case-studies/ebay/>
13. Використання Neo4j NASA [Електронний ресурс] – Режим доступу до ресурсу: <https://neo4j.com/blog/david-meza-chief-knowledge-architect-nasa/>
14. Використання Neo4j Monsanto [Електронний ресурс] – Режим доступу до ресурсу: <https://neo4j.com/blog/tim-williamson-data-scientist-monsanto/>

**Ісаков Андрій Васильович** – студент групи ІІІ-18б, факультет інформаційних технологій і комп’ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: [andrik.isakov@gmail.com](mailto:andrik.isakov@gmail.com)

**Ліміна Вероніка Юрївна** – студентка групи ІІІ-18б, факультет інформаційних технологій і комп’ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: [limina.nika@gmail.com](mailto:limina.nika@gmail.com)

**Романюк Оксана Володимирівна** – кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)

**Andrii Isakov** – student of group ІІІ-18b, Faculty of Information Technology and Computer Engineering, Vinnitsa National Technical University, Vinnitsa, e-mail: [andrik.isakov@gmail.com](mailto:andrik.isakov@gmail.com)

**Veronika Limina** – student of group ІІІ-18b, Faculty of Information Technology and Computer Engineering, Vinnitsa National Technical University, Vinnitsa, e-mail: [limina.nika@gmail.com](mailto:limina.nika@gmail.com)

**Romaniuk Oksana** – Candidate of Technical Sciences, Associate Professor of Software Engineering department, Vinnitsa National Technical University, Vinnitsa, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)