ВІСНИК ПРИАЗОВСЬКОГО ДЕРЖАВНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
2025р. Серія: Технічні науки Вип. 50
p-ISSN: 2225-6733; e-ISSN: 2519-271X

УДК 004.65:004.45 DOI: 10.31498/2225-6733.50.2025.336268

ПРИКЛАДИ ПРОЕКТУВАННЯ ДОКУМЕНТООРІЄНТОВАНОЇ БАЗИ ДАНИХ В MONGODB

Єлфімов Д.С. *бакалавр, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, e-mail: yelfimov_d_s@students.pstu.edu;*

Воротнікова З.Є. *канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: https://orcid.org/0000-0002-0746-5981, e-mail: vorotnikova_z_j@pstu.edu*

У статті детально досліджуються особливості та переваги використання документно-орієнтованих баз даних (NoSQL) як сучасної альтернативи традиційним реляційним системам управління даними. У контексті швидкого розвитку сучасної цифрової економіки та зростаючої необхідності у гнучких, масштабованих і високопродуктивних рішеннях, автори аналізують можливості застосування NoSQL баз даних у різних сферах діяльності. Це зокрема включає веб- та мобільні додатки, системи управління контентом, інтернет-магазини та інші сфери, де швидкість та масштабованість є критичними факторами. Проведено глибокий порівняльний аналіз реляційних та документоорієнтованих моделей зберігання даних, щоб показати їхні ключові відмінності та переваги. Для кожної моделі наведені приклади практичного застосування та особливості, що визначають їхню відповідність різним задачам. Особливу увагу приділено прикладам проєктування та реалізації баз даних для конкретних платформ, таких як блог-платформи та інтернет-магазини, з використанням MongoDB. У цих прикладах підкреслюється гнучкість структури даних, висока швидкість доступу до інформації та можливість масштабування у відповідь на зростаючий обсяг користувацьких запитів. Автори аналізують переваги кожної моделі, а також обговорюють її обмеження, підкреслюючи важливість правильного підходу до вибору архітектури бази даних залежно від характеру та вимог конкретної задачі. Загалом, стаття має на меті популяризацію сучасних технологій зберігання даних і підкреслює актуальність дослідження та впровадження NoSQL баз даних для підвищення ефективності, гнучкості та адаптивності інформаційних систем у сучасну епоху цифрових технологій. Вона сприяє розумінню того, як правильний вибір архітектури бази даних може позитивно вплинути на швидкість розробки, масштабованість та довгострокову стабільність інформаційних систем.

Ключові слова: *СКБД, RDBMS, NOSQL, JSON, JOIN, MongoDB, документоорієнтовані бази даних, блог-платформа, інтернет-магазин.*

Постановка проблеми

Для забезпечення конкурентоспроможності в сучасній цифровій економіці, орієнтованій на досвід користувача, підприємства змушені швидко впроваджувати інноваційні рішення, пов'язані з розробкою веб-, мобільних та IoT-додатків [1, 2]. Такий підхід вимагає прискорення процесів розробки та підвищення їхньої гнучкості, оскільки нові системи розвиваються значно швидше за старіючі традиційні рішення, такі як ERP. У цьому контексті реляційні бази даних виявляються обмеженими через свою фіксовану модель даних, що перешкоджає гнучкій адаптації. В якості альтернативи, бази даних типу NoSQL (Not Only SQL) дозволяють бізнесу простіше розробляти і масштабувати системи, забезпечуючи високу продуктивність і доступність, що особливо важливо в умовах динамічної цифрової економіки.

Безліч організацій вже використовують технології NoSQL, що спочатку застосовувалися для кешування або малих проектів, бази NoSQL поступово інтегрувалися в ключові програми і сьогодні виступають у ролі основи для розробки програмного забезпечення.

Вибір типу бази даних є важливим етапом в проектуванні інформаційних систем. Від нього залежить ефективність доступу до даних, масштабованість і безпека. Неправильний вибір може призвести до ускладнень, підвищених витрат на підтримку та погіршення користувальницького досвіду.

Реляційні бази даних широко застосовуються в сценаріях зі стабільною структурою даних і з високими вимогами щодо зв'язків, наприклад, в системах управління клієнтськими даними (CRM), фінансових додатках та системах обліку, де важлива чітка організація інформації про клієнтів, замовлення та продукти, а також виконання складних аналітичних запитів. У той же час, нереляційні бази даних ефективні в проектах з динамічними вимогами та великими обсягами даних, таких як веб-додатки, системи управління вмістом, соціальні мережі та хмарні сервіси, де структура даних має бути гнучкою та адаптованою до змін [2, 3].

Технологія NoSQL дозволяє зберігати дані у форматі документів JSON, що забезпечує гнучкість та масштабованість, а також підвищує швидкість реагування на потреби сучасного бізнесу. Термін «NoSQL» не означає відмову від SQL, а підкреслює можливість роботи з даними без використання традиційної мови запитів SQL, а також поєднання гнучкості формату JSON з потужністю SQL для поєднання переваг обох технологій.

Вивчення, популяризація та впровадження технологій NoSQL є актуальними завданнями підвищення ефективності розробки та експлуатації сучасних інформаційних систем.

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Аналіз останніх досліджень та публікацій

NoSQL бази даних використовуються для високонавантажених, розподілених та динамічних систем, де важлива гнучкість та швидкість. Розуміння специфіки кожної технології допомагає вибрати оптимальне рішення для різних програм.

До типів баз даних NoSQL належать документоорієнтовані БД (MongoDB, CouchDB, Couchbase, RavenDB, MarkLogic, ArangoDB, Elasticsearch), БД типу «ключ – значення» (Redis, Riak, Amazon DynamoDB, Aerospike, LevelDB, RocksDB, Berkeley DB, Memcached), колонкові БД (ClickHouse, Apache Cassandra, Amazon Redshift, Google BigQuery, Vertica, ClickHouse, Druid, MonetDB, Kdb+), часові БД (TimescaleDB, InfluxDB, OpenTSDB, Prometheus, KDB+/q, VictoriaMetrics, Graphite, QuestDB) та графові БД (Neo4j, JanusGraph, Amazon Neptune, OrientDB, TigerGraph, Blazegraph, Virtuoso, SAP HANA Graph). У статті [3] пропонується широкий огляд та класифікація статей на основі різних факторів, таких як мотиви зберігання великих даних, методи NoSQL, що використовуються для зберігання великих даних, та значущі системи великих даних у різних галузях.

Бази даних веб-сайтів та інтернет-магазинів були одними з перших, що перейшли на NoSQL-структуру. Це пов'язано з необхідністю масштабування та обробки великих обсягів неструктурованих даних, притаманних таким додаткам. Зокрема, платформи Amazon Dynamo (2007) та Cassandra (відкритий вихідний код Facebook, 2008) стали знаковими прикладами використання NoSQL у цьому контексті [4].

Проведені в статті [5] дослідження публікацій за останні 10 років показали, що документоорієнтовані та графові бази даних, особливо MongoDB, Cassandra та Neo4j, є найбільш вивченими, а підхід до навчання на основі проектів є найпоширенішим методом навчання.

У базах даних типу «ключ-значення» дані представлені у вигляді пар: унікальний ключ і пов'язане з ним значення. Як ключі, так і значення можуть бути будь-якими об'єктами – як простими, так і складними. Бази даних, орієнтовані на документи, також використовують пари «ключ-значення», але ключами можуть бути лише рядки, а значеннями – документи, закодовані у форматах типу JSON або XML, а також можуть включати PDF-файли, зображення або текстові файли. Це особливо зручно для роботи зі складними та ієрархічними структурами, де кожен документ може мати унікальні поля та вкладені дані. Вони широко застосовуються в системах управління контентом, мультимедійних платформах, зберіганні та аналізі великих обсягів напівструктурованих даних, а також у системах корпоративних знань та документації [3].

У статті [6] досліджується можливість документоорієнтованої бази даних з організації вихідних даних безпілотних літальних апаратів, таких як зображення та хмари точок. Документоорієнтованій структурі бази даних віддано перевагу порівняно з релаційною, оскільки вони здатні обробляти петабайти даних складної структури.

Порівняльна оцінка придатності реляційних баз даних (RDBMS) та документоорієнтованих баз даних (NoSQL) для додатків FinTech проводиться у статті [7]. Аналізуючи ключові показники, такі як продуктивність, масштабованість, узгодженість даних та експлуатаційна ефективність, це дослідження дає дієві відомості про сильні сторони та недоліки кожної архітектури бази даних. Дослідження оцінює реальні сценарії у контексті FinTech, включаючи управління транзакціями, зберігання даних клієнтів та аналітику.

У статті [8] розглядаються три підходи до адаптації багатовимірної моделі сховища історичних даних до Документоорієнтованих структур даних. Зазвичай історичні дані є таблицями фактів, оточені розмірними таблицями. Традиційно такі сховища підтримуються, переважно, реляційними базами даних.

Ряд статей [9, 10] присвячені автоматичним методам перетворення структур даних та міграції даних між реляційними та документоорієнтованими моделями зберігання даних.

У сучасному цифровому середовищі швидкий розвиток веб- та мобільних додатків, систем IoT і хмарних сервісів ставить під сумнів ефективність традиційних реляційних баз даних, оскільки вони не завжди здатні забезпечити необхідну гнучкість, масштабованість і продуктивність для динамічних і об’ємних даних. Водночас, технології NoSQL, зокрема документоорієнтовані бази даних, набувають все більшої популярності у сфері розробки сучасних інформаційних систем. Вивчення та порівняльний аналіз цих технологій має важливе практичне значення для оптимізації архітектури програмних продуктів, підвищення їхньої ефективності та адаптивності до швидкозмінних вимог ринку.

Мета статті

Метою роботи є дослідження особливостей та переваг документоорієнтованих систем керування базами даних (СКБД) у контексті їх застосування для сучасних веб- та мобільних додатків, а також порівняльна оцінка з традиційними реляційними базами даних. Важливою ціллю є також ілюстрація процесу переходу від реляційної до документоорієнтованої моделі на прикладах реальних сценаріїв, таких як блог-платформа та інтернет-магазин, з метою демонстрації переваг і особливостей кожного підходу.

В рамках дослідження передбачено аналіз сучасних типів NoSQL баз даних, їх структурних особливостей, можливостей та обмежень; порівняльне вивчення моделей даних (реляційної та документної); розробку прикладів проектування баз даних для популярних сценаріїв; а також оцінку впливу вибору типу бази даних на продуктивність, масштабованість і зручність управління даними у реальних системах.

Виклад основного матеріалу

Для ілюстрації процесу проектування нереляційної бази даних було обрано дві поширені задачі, база даних для яких історично будувалася як реляційна: «блог» та «інтернет-магазин». Для розробки структури даних було обрано документоорієнтовану структуру [5].

Документоорієнтовані бази даних є системами зберігання даних, у яких кожен елемент – це структурований документ, що містить набір пар ключ-значення. Ці документи можуть бути записані у форматах, таких як JSON, BSON, XML та аналогічних, та групуються в колекції, забезпечуючи ефективну організацію та взаємозв'язок даних. На відміну від реляційних баз даних, де структура записів суворо фіксована, у документоорієнтованих системах відсутня схема даних, що дозволяє динамічно змінювати структуру окремих документів без необхідності глобальної модифікації бази. Кожному документу надається унікальний ідентифікатор, зазвичай рядкового типу, що забезпечує швидкий доступ та міждокументні посилання.

Однією з суттєвих переваг документоорієнтованої БД є гнучкість, що дозволяє легко додавати нові атрибути та розширювати структуру даних, а також масштабованість за рахунок горизонтального розподілу навантаження. Документні бази даних забезпечені потужним засобами індексації та складних запитів, що сприяє високій продуктивності при пошуку та фільтрації даних, а також підтримують багатомовні та мультимедійні сценарії за рахунок можливості зберігання різних локалізацій та медіа-даних усередині документів.

Однак документоорієнтовані моделі мають і обмеження: складності при реалізації міждокументних запитів, ризик надмірності даних та неефективність при роботі зі складними взаємозв'язками чи транзакціями.

Практичне застосування документних баз даних широко поширене в сферах управління контентом, каталогізації та інтернету речей (IoT). У системах керування контентом такі бази дозволяють легко вносити зміни та оновлювати окремі документи без переривання роботи системи. У каталогах, наприклад, для інтернет-комерції, вони забезпечують ефективне зберігання товарів із різноманітними атрибутами, де зміни окремих елементів не торкаються інших. В IoT-проектах вони дозволяють оперативно зберігати потокові дані з датчиків, не вимагаючи попередньої схеми та забезпечуючи масштабованість для обробки великих обсягів інформації.

До популярних систем цього типу відносяться MongoDB, CouchDB, Firebase Firestore та Amazon DocumentDB.

Для реалізації прикладів документоорієнтованої бази даних для завдань «блог» та «інтернет-магазин» була обрана СУБД MangoDB як найпопулярніша і найпростіша в освоєнні СУБД для документоорієнтованої бази даних [5].

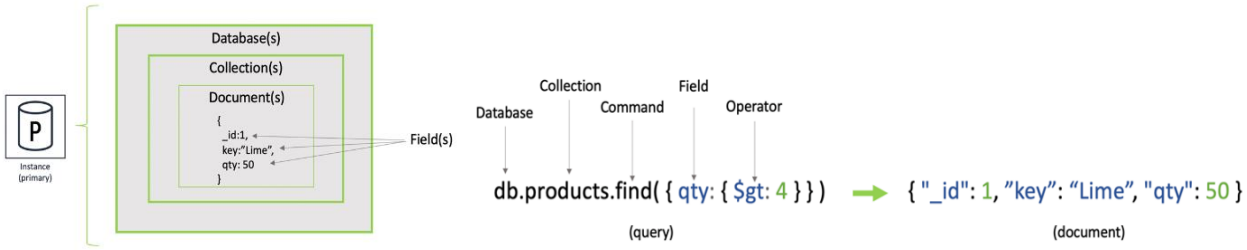


Рис. 1 – Структура документоорієнтованої бази даних

Таблиця 1

Порівняння реляційної та документної баз даних

РСКДБ	Система документних баз даних
Побудовано коло концепції про зв'язки	Зосереджена на даних, а не не зв’язках
Структурує данні в кортежі (строки)	Замість строк в документах є різноманітні характеристики
Визначає данні (утворює зв’язки) через обмеження та зовнішні ключі (залежна таблиця посилається на головну через її ідентифікатор)	Замість зовнішніх ключів зв’язки реалізовано через вкладені данні (в один документ може бути вложено інші документи, що формує зв’язок)
Для утворення зв’язків використовує мову DLL	Не потребує мови DLLдля утворення зв’язків
Забезпечує виключну узгодженість даних. У певних випадках її конче необхідно (наприклад, щоденні банківські операції)	Забезпечує узгодженість у підсумку (з періодом неузгодженості)

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Приклад 1. Блог-платформа

Блог-платформа – призначена для створення, публікації, перегляду та управління контентом у форматі блогу. Вона дозволяє авторам писати статті (пости), категоризувати їх, додавати теги, а читачам – переглядати ці статті, залишати коментарі та взаємодіяти з платформою. База даних використовується для наступних дій:

1. Зберігання контенту: облікових записів користувачів, публікацій, категорій, тегів, коментарів тощо.
2. Організація даних: необхідно чітко визначити зв'язки між різними типами даних (наприклад, кожен пост належить автору, може мати кілька категорій та тегів, до нього можуть бути залишені коментарі). Це полегшує пошук, фільтрацію та управління даними.
3. Забезпечення цілісності даних: використання обмежень (NOT NULL, UNIQUE, FOREIGN KEY) гарантує консистентність та валідність даних у базі.
4. Ефективні запити: система запитів повинна гнучко та ефективно отримувати необхідну інформацію з бази даних для відображення на веб-сайті, формування стрічок новин, пошуку за ключовими словами тощо.

Така структура бази даних використовується для реалізації бекенд-частини блог-платформ. Наприклад, персональні блоги, для ведення власних онлайн-щоденників; тематичні блоги – платформи (технології, подорожі, кулінарія); корпоративні блоги, для публікації новин, статей, експертних думок; новинні веб-сайти, для управління статтями та іншим контентом, тощо.

Історично для таких систем в якості сховища даних обирали реляційну базу даних (наприклад, MySQL, PostgreSQL, SQLite). Але з ростом якості та об’єму контенту, що зберігається та обробляється, збільшенням кількості користувачів, є сенс використовувати документоорієнтовану структуру бази даних. Такий підхід дозволить суттєво прискорити роботу блог-платформи та зняти проблеми, пов’зані з маштабуванням бази даних.

Технології RDBMS/SQL і NoSQL мають дуже різні підходи до представлення даних. У цій роботі зроблено спробу показати на прикладах, яким чином RDBMS/SQL, його функціонал і мова запитів може бути відображений в базі даних MongoDB.

Кожна база даних MongoDB складається з колекцій, аналогічних тому, як у реляційних системах управління базами даних (RDBMS) бази складаються з таблиць. Кожна колекція зберігає дані у вигляді документів, які по суті відповідають рядкам таблиць SQL. У той час як рядки містять дані у вигляді набору колонок, документи являють собою JSON-подібні структури (BSON у MongoDB). Аналогічно стовпцям SQL, поля в документах задають структуру даних [11].

На рис. 2 зображено базу даних MongoDB, що складається з двох колекцій «posts» та «users», призначених для зберігання інформації про пости та користувачів блог-платформи. Ця структура відповідає предметній області, представленій ER-моделлю на рисунку 3, де інформація про пости міститься в таблицях «Posts», «Comments», «Replies», «Metadata», «Tags», а інформація про користувачів – у таблицях «Users», «Stats» та «Roles».

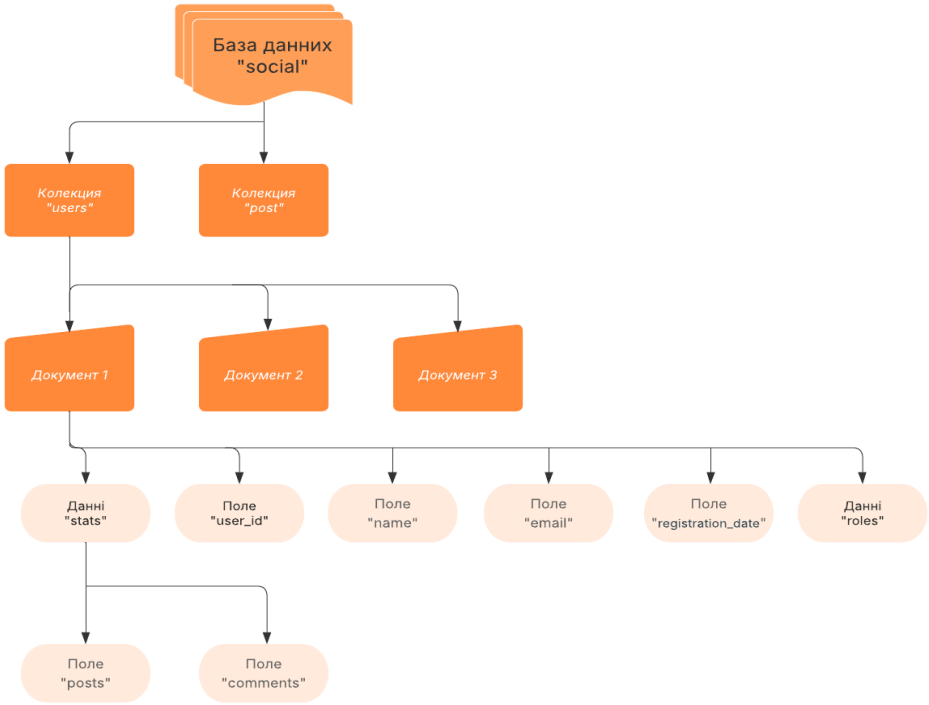


Рис. 2 – Структура даних в MongoDB

p-ISSN: 2225-6733; e-ISSN: 2519-271X

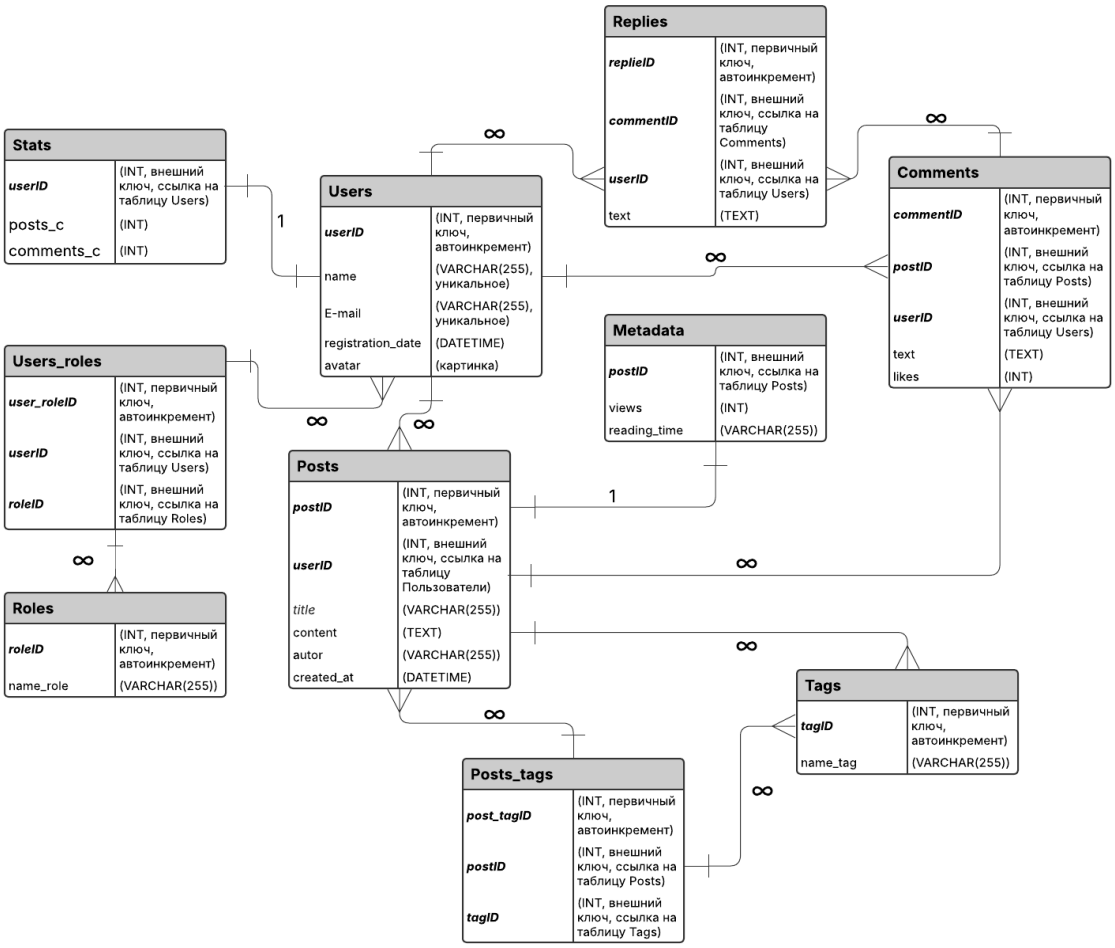
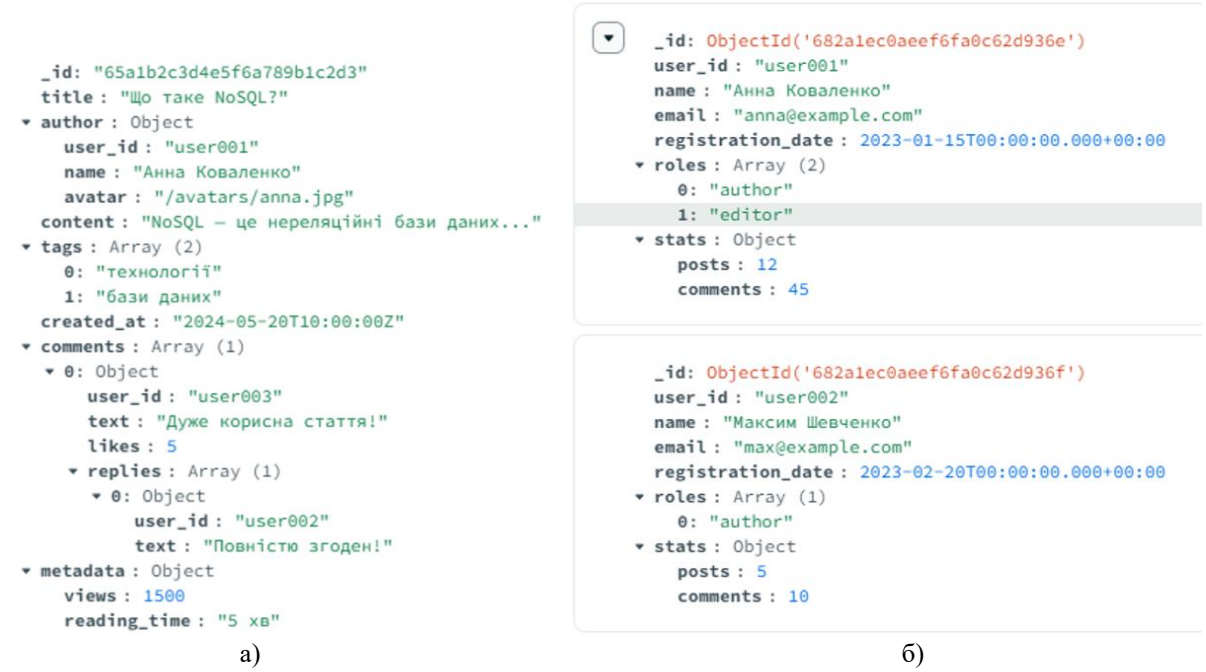


Рис. 3 – Реляційна схема даних для прикладу 1

На рисунку 4 наведено приклади документів (аналог рядків SQL), що містять кілька полів (аналог колонок). В одній колекції документи можуть мати різну структуру: один документ може містити п'ять полів, інший – сім. Наприклад, у першому документі (див. рисунок 4б) є поле «1», якого немає у другому документі. У разі реалізації подібної схеми у RDBMS довелося б додавати додаткову колонку, яка могла б залишатися порожньою у деяких рядків, що займало б зайве місце. Поля легко додавати або видаляти будь-коли, при цьому немає обмежень за типами даних. Наприклад, в одному документі поле може містити ціле число, а в іншому – масив.

Така модель динамічної схеми робить документоорієнтовані бази даних дуже гнучкими щодо проектування. Різні складні структури (наприклад, ієрархії, деревоподібні схеми), які вимагають кількох зв'язаних таблиць у RDBMS, зручно реалізуються за допомогою документів. Наприклад, пости, коментарі, лайки та інша пов'язана інформація можуть зберігатися в одному документі, а не розподілятися за кількома таблицями. Це дозволяє при необхідності швидко змінювати структуру документів і адаптувати їх під вимоги, що змінюються.

У реляційних базах даних зв'язки реалізуються через первинні та зовнішні ключі, а також за допомогою JOIN-запитів. Для нашого прикладу, щоб відтворити зв’язки багато-до-багатьох у реляційній базі даних, потрібно створити навіть дві допоміжні таблиці. У MongoDB прямих аналогів зв'язків немає, але їх можна реалізувати за допомогою посилань на інші документи чи вкладених документів. Зазвичай у MongoDB використовується автоматично створюване поле _id як унікальний ідентифікатор документа. У нашому прикладі колекції «posts» та «users» пов'язані через поле user_id, яке містить ідентифікатор користувача. Важливо пам'ятати, що в MongoDB зв'язки та операції з ними вимагають ретельного відстеження, тому що немає обмежень на зовнішні ключі. Якщо вставити посилання на неіснуючий документ, то помилка фіксуватися системою не буде, на відміну від SQL, де це викликає помилку цілісності даних.



а) б)

Рис. 4 – Приклади документів створених в MongoDB: а) документ з коллекції «posts»; б) документи з колекції «users»

Використання вкладених документів (див. рисунок 5) дозволяє створювати складні та ієрархічні структури всередині одного документа, використовуючи типи даних Array та Object.

Рис. 5 – Приклади вкладених документів

Вибір між використанням посилання або вбудованого вкладеного документа залежить від ситуації. Якщо передбачається, що вкладені дані можуть збільшитися з часом, краще використовувати посилання, щоб уникнути занадто великого розміру документа. Якщо кількість вкладених елементів обмежена, зручно використовувати вбудовані документи.

Бази даних документів надають мову запитів або API, що дозволяють працювати з даними, що зберігаються. У MongoDB можна виконувати операції створення, читання, оновлення та видалення баз даних, колекцій та документів, а також окремих їх полів.

На рисунку 6, представлено приклади виконання запитів: а) додавання посту до колекції posts; б) пошук інформації за умовою (про пост за назвою «Новий пост»); в) оновлення даних у посту (дані значення поля likes).

а) б) в)

Рис. 6 – Приклади виконання запитів в MongoDB: а) додавання посту, команда db.MyCollection.insertOne(); б) пошук інформації за умовою; в) оновлення даних

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Структура ДООБД починається з кореневого вузла, тоді листові вузли містять кінцеві дані. Пошук здійснюється за принципом «ключ-значення», тому не підходить для складних запитів через складність пошуку між колекціями та можливі проблеми з узгодженістю даних.

Вибір конкретної моделі залежить від обсягу, складності та масштабованості платформи. Реляційні бази даних мають структуру таблиць із зв'язками між ними, що забезпечує цілісність і складні запити, тоді як документоорієнтовані бази даних пропонують більшу гнучкість у структурі даних, динамічне додавання полів та зручність для ієрархічних та складних структур. MongoDB особливо підходить для швидкої адаптації та збереження складних об'єктів у вигляді документів, що робить її ефективним рішенням для сучасних блог-систем.

Приклад 2. Інтернет-магазин

Документоорієнтована модель MongoDB ідеально підходить для інтернет-магазинів, де:

– товари мають багато варіацій (розміри, кольори);

– потрібна швидка робота з даними (пошук, фільтрація);

– немає необхідності в складних JOIN-запитах.

Якщо потрібно реалізувати складні зв’язки (наприклад, замовлення з багатьма товарами), можна використовувати референційну модель (посилання на інші документи).

У моделі MongoDB кожен товар може бути представлений у вигляді єдиного документу, який містить усю необхідну інформацію, включаючи варіації (кольори, розміри, характеристики тощо). Це дозволяє уникнути складнощів, пов’язаних із реляційними зв’язками (JOIN-запити), і забезпечує швидкий доступ до даних. Усі дані про товар (включаючи наявність на складі) завантажуються одним запитом. Також MongoDB дозволяє швидко фільтрувати, сортувати та групувати товари (наприклад, знайти всі футболки розміру «M»).

На рисунках 7, 8 зображено структури реляційної бази даних (рис. 7) та даних MongoDB (рис. 8).

На рисунку 9 наведено приклади документів, що можна використовувати для формування структури бази даних для інтернет-магазину.

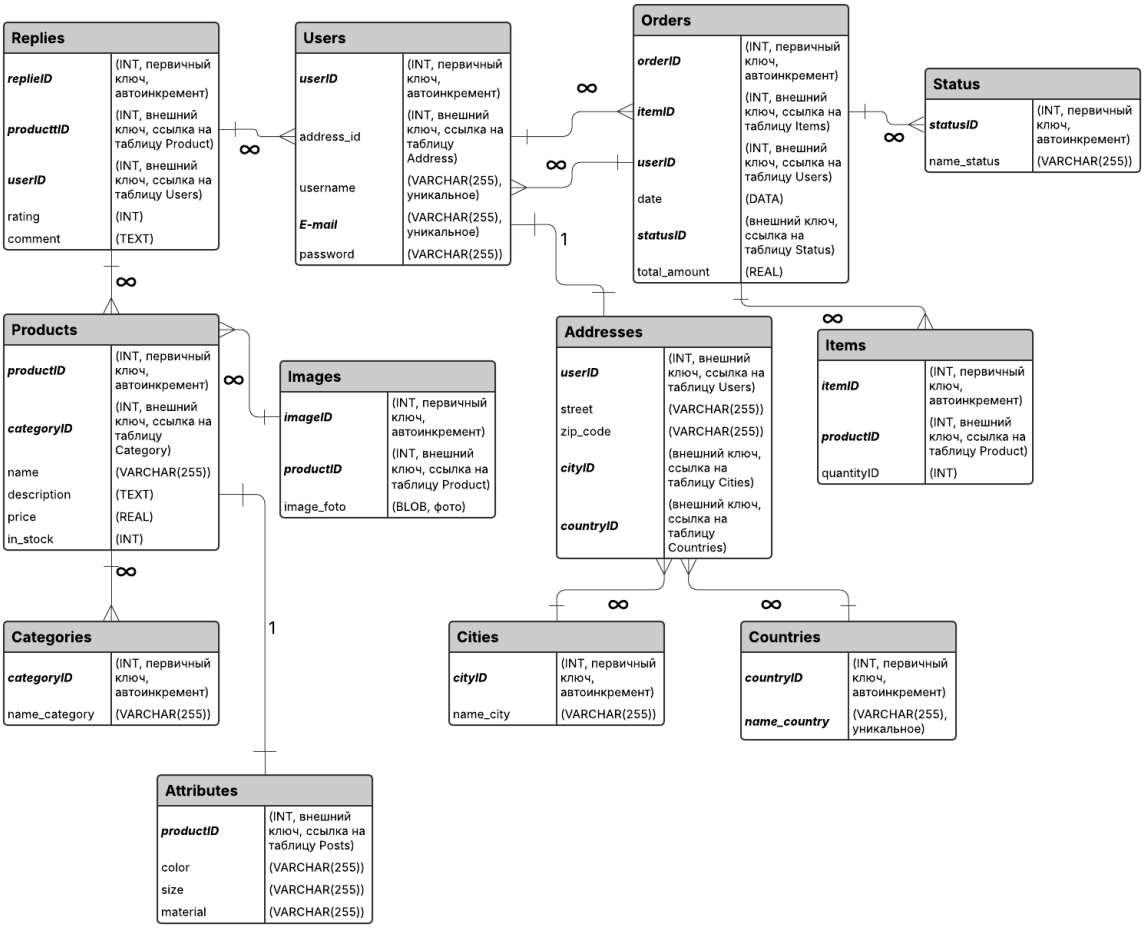


Рис. 7 – Відповідність структур даних для прикладу 2: реляційна схема даних

p-ISSN: 2225-6733; e-ISSN: 2519-271X

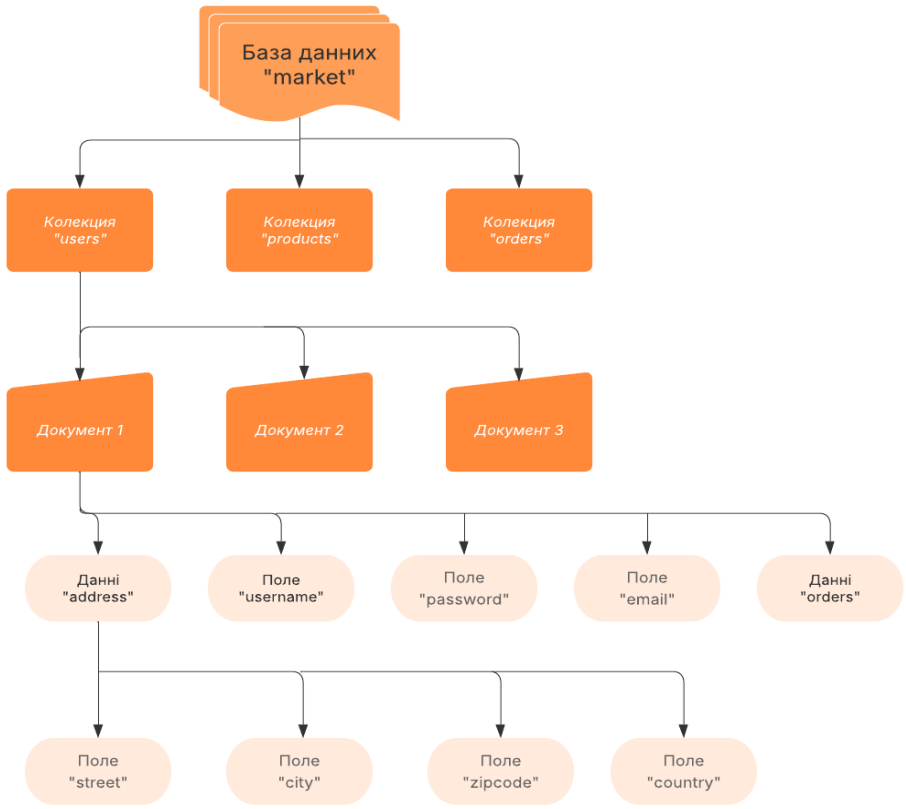


Рис. 8 – Відповідність структур даних для прикладу 2: структура даних в MongoDB

а) б)

Рис. 9 – Приклади документів, створених в MongoDB для інтернет-магазину: а) колекція orders; б) колекція users

p-ISSN: 2225-6733; e-ISSN: 2519-271X

а) б) в)

Рис. 10 – Приклади виконання запитів в MongoDB: а) пошук користувачів у місті Львів; б) продукти у категорії «Одяг»; в) оновлення даних

Приклади запитів, які можна поставити до бази даних «інтернет-магазин», представлено на рисунку 10.

Для швидкого пошуку товарів та фільтрації за різними параметрами використовують індекси, що значно підвищує продуктивність запитів. Крім того, MongoDB підтримує масштабування шардуванням, що дозволяє рівномірно розподіляти дані кількома серверами. Завдяки горизонтальному масштабуванню, MongoDB може обробляти великі обсяги даних та високий трафік, що важливо для інтернет-магазинів із зростаючою відвідуваністю. Також варто відзначити вбудовану реплікацію та автоматичне відновлення даних, що забезпечує високу надійність системи. Важливою особливістю є можливість зберігання зображень та мультимедійних файлів у базі даних за допомогою GridFS. В цілому, MongoDB дозволяє створювати гнучкі та масштабовані рішення для інтернет-магазинів, забезпечуючи швидку роботу та зручність управління даними.

Перехід від реляційної структури даних до документоорієнтованої для сайту (блогу, або інтернет-магазину) спричиняє зміну способу зберігання та організації інформації. Такий перехід має свої переваги та недоліки.

Документоорієнтовані бази краще підходять для проектів, де структура даних може змінюватися з часом, тобто для сайтів, що потребують горизонтального масштабування (розподіл навантаження на кілька серверів). Перевага розробки за рахунок можливості зберігати складні структури даних в одному документі, уникаючи складних JOIN операцій, які часто використовуються в реляційних базах, також приваблива для розробників сайтів. У деяких випадках, для певних типів запитів документоорієнтовані бази даних можуть забезпечувати кращу продуктивність, особливо при роботі з даними, які не потребують складної структури зв'язків.

Перенесення існуючих даних з реляційної бази до документоорієнтованої може вимагати значних зусиль та часу. Перехід вимагає перегляду архітектури програми, щоб адаптувати її до нових принципів роботи з даними. Так само, відсутність суворої структури даних у документоорієнтованих базах може призвести до проблем узгодженості даних, якщо не буде розроблено відповідну систему валідації та управління даними.

Рішення про перехід має ухвалюватися на основі конкретних вимог проекту. Якщо сайт має складну структуру даних, вимагає високої гнучкості та масштабованості, а також не вимагає суворого зв'язку між даними, то перехід на документоорієнтовану базу даних може бути виправданий. Якщо ж сайт має строгу структуру даних, вимагає високого ступеня узгодженості даних і не має проблем із масштабуванням, то перехід може виявитися недоцільним

Висновки

У рамках цього дослідження створено глибоке розуміння особливостей та можливостей документоорієнтованих баз даних, проілюстровано їхню реалізацію на прикладах популярних задач, а також сформульовано рекомендації щодо їхнього застосування у сучасних інформаційних системах. Для цього було виконано наступні дослідження:

– аналіз сучасних тенденцій та обґрунтування актуальності використання NoSQL баз даних у сучасних інформаційних системах для підвищення гнучкості, масштабованості та продуктивності;

– вивчено особливості документоорієнтованих баз даних, зокрема MongoDB, як ефективного інструменту для зберігання складних та ієрархічних структур даних у розробці веб-додатків;

– порівняно реляційні бази даних з документоорієнтованими моделями, показано переваги та недоліки кожної архітектури, а також можливості їхнього поєднання та переходу між ними;

– на основі прикладів для двох типових завдань («блог-платформа» та «інтернет-магазин») розроблено структури даних у документоорієнтованій базі MongoDB, ілюстровано їхню реалізацію, в тому числі через створення колекцій і зразків документів;

– продемонстровано, як можна моделювати зв'язки та реалізовувати запити в MongoDB, порівняно з реляційною моделлю, забезпечуючи розуміння принципів роботи та особливостей кожної підходу;

– проведено аналіз переваг та викликів переходу від реляційної до документоорієнтованої моделі, враховуючи питання гнучкості, масштабованості, цілісності даних та архітектурних особливостей;

– надано рекомендації щодо застосування й вибору відповідної моделі баз даних залежно від специфіки проекту, а також можливості автоматизації переходу між цими моделями.

Перелік використаних джерел

[1] NoSQL: Future of BigData Analytics Characteristics and Comparison with RDBMS / M. Arshad et al. *The Effect of Information Technology on Business and Marketing Intelligence Systems. Studies in Computational Intelligence*; edited by M. Alshurideh et al. 2023. Vol. 1056. Pp. 1927-1951. DOI: https://doi.org/10.1007/978-3-031-12382-5_106.

[2] A state of art survey for Big Data processing and NoSQL database architecture / Ali A., Naeem S., Anam S., Ahmed M. *International Journal of Computing and Digital Systems*. 2023. Vol. 14, no. 1. Pp. 1-1. DOI: http://dx.doi.org/10.12785/ijcds/140124.

[3] Faridoon A., Imran M. Big Data Storage Tools Using NoSQL Databases and Their Applications in Various Domains: A Systematic Review. *Computing & Informatics*. 2021. Vol. 40, no. 3. Pp. 489-521. DOI: https://doi.org/10.31577/cai_2021_3_489.

[4] Kraiem M.B., Feki J. Building a Data Warehouse for Social Media: Review and Comparison. *Computación y Sistemas*. 2024. Vol. 28, no. 1. Pp. 19-39. DOI: https://doi.org/10.13053/cys-28-1-4677.

[5] Tripathi N. NoSQL database education: A review of models, tools and teaching methods. *Journal of Systems and Software*. 2025. Vol. 226. Article 112391. DOI: https://doi.org/10.1016/j.jss.2025.112391.

[6] Document-Oriented Data Organization for Unmanned Aerial Vehicle Outputs / S. Azri et al. *Journal of Advances in Information Technology*. 2021. Vol. 12, no. 4. DOI: https://doi.org/10.12720/jait.12.4.267-278.

[7] Ionescu S.A., Radu A.O. Assessment and Integration of Relational Databases, Big Data, and Cloud Computing in Financial Institutions: Performance Comparison. *International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Craiova, Romania, 04-06 September 2024. Pp. 1-7. DOI: https://doi.org/10.1109/INISTA62901.2024.10683852.

[8] Matias R., Piedade M.B. Multidimensional Models Supported by Document-Oriented Databases. *International Conference on Hybrid Artificial Intelligence Systems*, Salamanca, Spain, 5-7 September 2023. Pp. 156-167. DOI: https://doi.org/10.1007/978-3-031-40725-3_14.

[9] Tandya W., Azizah F.N. Migration of Relational Database to NoSQL Document-Oriented Database. *IEEE International Conference on Data and Software Engineering (ICoDSE)*, Toba, Indonesia, 07-08 September 2023. Pp. 180-185. DOI: https://doi.org/10.1109/ICoDSE59534.2023.10291584.

[10] Peretiatko M., Shirokopetleva M., Lesna N. Research of methods to support data migration between relational and document data storage models. *Innovative technologies and scientific solutions for industries*. 2022. Vol. 2, no. 20. Pp. 64-74. DOI: https://doi.org/10.30837/ITSSI.2022.20.064.

[11] MongoDB Home Page. URL: https://www.mongodb.com (дата звернення: 26.11.2024).

EXAMPLES OF DESIGNING A DOCUMENT-ORIENTED DATABASE IN MONGODB

Yelfimov D. *Bachelor's student, SHEI «Priazovskyi state technical university», Dnipro, e-mail: yelfimov_d_s@students.pstu.edu;*

Vorotnikova Z. *PhD (Engineering), associate professor, SHEI «Priazovskyi state technical university», Dnipro, ORCID: https://orcid.org/0000-0002-0746-5981, e-mail: vorotnikova_z_j@pstu.edu*

The article provides a detailed exploration of the features and advantages of using document-oriented databases (NoSQL) as a modern alternative to traditional relational database management systems. In the context of the rapid development of the modern digital economy and the increasing need for flexible, scalable, and high-performance solutions, the authors analyze the possibilities of applying NoSQL databases across various fields. This includes web and mobile applications, content management systems, online stores, and other areas where speed and scalability are critical factors. A comprehensive comparative analysis is conducted between relational and document-oriented data storage models to highlight their key differences and benefits. For each model, practical application examples and distinctive features that determine their suitability for different tasks are presented. Special attention is given to examples of designing and implementing databases for specific platforms, such as blog platforms and online stores, using MongoDB. These examples emphasize

the flexibility of data structures, high-speed data access, and the ability to scale in response to growing user demands. The authors analyze the advantages of each model, as well as discuss their limitations, underscoring the importance of choosing the appropriate database architecture based on the nature and requirements of a particular task. Overall, the article aims to promote modern data storage technologies and highlights the relevance of researching and implementing NoSQL databases to enhance the efficiency, flexibility, and adaptability of information systems in the current era of digital technologies. It facilitates understanding of how the correct choice of database architecture can positively impact development speed, scalability, and the long-term stability of information systems.
***Keywords**: DBMS, RDBMS, NOSQL, JSON, JOIN, MongoDB, document-oriented databases, blog platform, online store.*

References

[1] M. Arshad, M. Brohi, T. Soomro, T. Ghazal, H. Alzoubi, and M. Alshurideh, «NoSQL: Future of Big-Data Analytics Characteristics and Comparison with RDBMS», in *The Effect of Information Technology on Business and Marketing Intelligence Systems. Studies in Computational Intelligence*, M. Alshurideh, B.H. Al Kurdi, R. Masa'deh, H.M. Alzoubi, S. Salloum, Eds., 2023, vol 1056, Springer, Cham, pp. 1927-1951. **doi: 10.1007/978-3-031-12382-5_106**.

[2] A. Ali, S. Naeem, S. Anam, and M. Ahmed, «A state of art survey for Big Data processing and NoSQL database architecture», *International Journal of Computing and Digital Systems*, vol. 14. no. 1. pp. 1-1, 2023. **doi: 10.12785/ijcds/140124**.

[3] A. Faridoon, and M. Imran, «Big Data Storage Tools Using NoSQL Databases and Their Applications in Various Domains: A Systematic Review», *Computing & Informatics*, vol. 40, no. 3, pp. 489-521, 2021. **doi: 10.31577/cai_2021_3_489**.

[4] M.B. Kraiem, and J. Feki, «Building a Data Warehouse for Social Media: Review and Comparison», *Computación y Sistemas*, vol. 28, no. 1, pp. 19-39, 2024. **doi: 10.13053/cys-28-1-4677**.

[5] N. Tripathi, «NoSQL database education: A review of models, tools and teaching methods», *Journal of Systems and Software*, vol. 226, article 112391, 2025. **doi: 10.1016/j.jss.2025.112391**.

[6] S. Azri, U. Ujang, W. Embong, M. Cuetara, and G. Miguel, «Document-Oriented Data Organization for Unmanned Aerial Vehicle Outputs», *Journal of Advances in Information Technology*, vol. 12. no. 4, 2021. **doi: 10.12720/jait.12.4.267-278**.

[7] S.A. Ionescu, and A.O. Radu, «Assessment and Integration of Relational Databases, Big Data, and Cloud Computing in Financial Institutions: Performance Comparison», in Proc. of the Int. Conf. on INnovations in Intelligent SysTems and Applications (INISTA), Craiova, Romania, 2024, pp. 1-7. **doi: 10.1109/INISTA62901.2024.10683852**.

[8] R. Matias, and M.B. Piedade, «Multidimensional Models Supported by Document-Oriented Databases», in Proc. of the Int. Conf. on Hybrid Artificial Intelligence Systems, Salamanca, Spain, 2023, pp. 156-167, 2023. **doi: 10.1007/978-3-031-40725-3_14**.

[9] W. Tandya, and F.N. Azizah, «Migration of Relational Database to NoSQL Document-Oriented Database», in Proc. of the IEEE Int. Conf. on Data and Software Engineering (ICoDSE), Toba, Indonesia, 2023, pp. 180-185, 2023. **doi: 10.1109/ICoDSE59534.2023.10291584**.

[10] M. Peretiatko, M. Shirokopetleva, and N. Lesna, «Research of methods to support data migration between relational and document data storage models», *Innovative technologies and scientific solutions for industries*, vol. 2, no. 20, pp. 64-74, 2022. **doi: 10.30837/ITSSI.2022.20.064**.

[11] MongoDB Home Page. [Online]. Available: https://www.mongodb.com. Accessed on: November 26, 2024.

Стаття надійшла 25.03.2025
Стаття прийнята 18.04.2025
Стаття опублікована 30.06.2025

Цитуйте цю статтю як: Єлфімов Д.С., Воротнікова З.Є. Приклади проектування документоорієнтованої бази даних в MongoDB. *Вісник Приазовського державного технічного університету. Серія: Технічні науки*. 2025. Вип. 50. С. 50–60. DOI: https://doi.org/10.31498/2225-6733.50.2025.336268.