УДК 004.652.2:004.65

ПОРІВНЯЛЬНИЙ АНАЛІЗ АРХІТЕКТУР SQL ТА NOSQL БАЗ ДАНИХ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Антоненко П. П.[1]; Нестеров Ю. М.[2]
[1, 2]Національний університет «Одеська політехніка»
Україна, Одеса
[1]pavlo.antonenko@gmail.com; [2]nesterov@op.edu.ua

Анотація. *У роботі проведено порівняльний аналіз архітектур реляційних та нереляційних баз даних в контексті сучасних інформаційних систем. Розглянуто структурні відмінності, моделі забезпечення цілісності, масштабованість, продуктивність та сфери ефективного застосування обох підходів. Виявлено переваги і недоліки кожної архітектури залежно від типу даних і навантаження.*
Ключові слова: *SQL; NoSQL; бази даних; інформаційні системи; архітектура.*

Вступна частина. У сучасному інформаційному суспільстві обсяг даних, що створюється та обробляється, зростає експоненціально. З розвитком технологій Big Data, хмарних обчислень та інтернету речей (IoT) з'являється потреба у більш гнучких і масштабованих системах зберігання інформації [1]. Традиційні реляційні бази даних (SQL), які десятиліттями залишалися стандартом для роботи зі структурованими даними, уже не завжди здатні забезпечити необхідну продуктивність і масштабованість у сучасних умовах [2]. На зміну класичним підходам приходять нереляційні бази даних (NoSQL), які пропонують іншу архітектуру та підхід до обробки даних – без жорстких схем і з фокусом на розподілене зберігання [3].

Актуальність цієї теми зумовлена потребою глибше зрозуміти відмінності між SQL і NoSQL системами, визначити їх переваги, недоліки та можливості використання в різних типах інформаційних систем. Порівняння архітектур цих двох підходів дозволяє не лише оцінити ефективність їх роботи в різних умовах, але й зробити висновки щодо доцільності використання кожної технології залежно від типу даних, обсягів інформації та вимог до швидкодії.

Метою дослідження є проведення аналітичного порівняння архітектур SQL та NoSQL баз даних з метою визначення їх структурних, функціональних і продуктивних відмінностей.

Основна частина. Реляційна парадигма базується на зберіганні даних у вигляді таблиць, пов'язаних між собою ключами, та використанні мови структурованих запитів SQL для маніпулювання даними. У свою чергу, нереляційна (NoSQL) парадигма орієнтована на роботу з неструктурованими або напівструктурованими даними, забезпечуючи високу масштабованість і гнучкість за рахунок відмови від жорсткої схеми даних. Порівняння цих двох підходів дозволяє глибше зрозуміти сучасні тенденції розвитку систем зберігання інформації.

Реляційні бази даних характеризуються чіткою структурою, де дані зберігаються у вигляді таблиць (відношень), а зв'язки між ними визначаються первинними та зовнішніми ключами. Така модель забезпечує логічну цілісність, узгодженість і надійність даних, що зробило її стандартом де-факто у сфері інформаційних технологій.

Однак із розвитком інтернет-технологій, появою соціальних мереж і стрімким зростанням обсягів даних виникла потреба у нових моделях зберігання. Це спричинило появу нереляційних (NoSQL) систем, які відмовилися від жорсткої табличної структури на

користь більш гнучких способів представлення даних: документів, графів, пар “ключ–значення” або колонкових сховищ. Ці системи орієнтовані на горизонтальне масштабування та високу продуктивність при роботі з великими обсягами неструктурованої інформації.

Принципи побудови архітектур баз даних передбачають поділ на логічний, фізичний і концептуальний рівні. Логічний рівень визначає, як дані представлені користувачу, фізичний – як вони зберігаються у пам’яті чи на диску, а концептуальний описує загальну модель даних і зв’язків між ними. У реляційних системах акцент робиться на логічній структурі та узгодженості даних, тоді як у NoSQL архітектура орієнтована на гнучкість і швидкість обробки.

За моделями зберігання даних системи управління базами даних (СУБД) класифікуються на кілька основних типів:

– реляційні (SQL) – базуються на таблицях і підтримують транзакційність за принципом ACID;

– документоорієнтовані – зберігають дані у вигляді JSON або XML-документів;

– колонкові – оптимізовані для аналітичних обчислень і великих наборів даних;

– графові – зберігають інформацію у вигляді вузлів і зв’язків, що зручно для соціальних мереж або рекомендаційних систем;

– ключ-значення – прості й високошвидкісні, використовуються для кешування або швидкого доступу до даних.

Таким чином, архітектура баз даних еволюціонувала від суворо структурованих реляційних моделей до гнучких, масштабованих нереляційних систем. Обидва підходи мають свої переваги та обмеження, а вибір між ними залежить від конкретних потреб системи, характеру даних і вимог до швидкодії та масштабованості.

Реляційна архітектура баз даних є однією з найпоширеніших і найстабільніших моделей зберігання інформації, яка зберегла свою актуальність упродовж кількох десятиліть. Основою цієї моделі є представлення даних у вигляді таблиць (відношень), між якими встановлюються логічні зв’язки. Такий підхід забезпечує високу структурованість, узгодженість і передбачуваність даних.

Типова архітектура SQL-СУБД складається з кількох ключових компонентів. Першим є сховище даних (storage engine) – фізичний рівень системи, який відповідає за розміщення, збереження та доступ до інформації на дисках. Другим важливим елементом є оптимізатор запитів (query optimizer) – програмний модуль, який аналізує SQL-запити, обирає найефективнішу стратегію їх виконання і формує план доступу до даних. Транзакційний менеджер (transaction manager) забезпечує цілісність і узгодженість операцій, виконуваних у системі. Він контролює початок, фіксацію або відкат транзакцій, що особливо важливо при одночасній роботі багатьох користувачів. Крім того, до архітектури входять менеджер буферів, який працює з кешуванням даних у пам’яті, та механізм журналювання (logging system), що дозволяє відновити систему після збоїв.

Серед основних переваг реляційного підходу можна відзначити високу цілісність даних, наявність стандартизованої мови SQL, гнучкі можливості запитів, підтримку транзакцій та широке поширення серед різних платформ. До обмежень належить складність масштабування, значні витрати ресурсів при роботі з великими обсягами даних та обмежена гнучкість структури через жорстку схему.

Найвідомішими представниками реляційних СУБД є MySQL, PostgreSQL, Oracle Database та Microsoft SQL Server, які стали основою для більшості сучасних корпоративних систем і веб-застосунків.

З розвитком інтернет-технологій, зростанням обсягів неструктурованих даних і вимог до масштабованості виникла потреба у нових моделях зберігання інформації, що не залежать від жорсткої табличної структури. Так з'явився клас нереляційних баз даних (NoSQL – Not Only SQL), які пропонують альтернативний підхід до організації даних і побудови архітектури.

NoSQL системи характеризуються гнучкою структурою даних, можливістю горизонтального масштабування та високою продуктивністю при обробці великих масивів інформації. Вони не потребують фіксованої схеми таблиць, що дозволяє адаптувати модель даних до змін бізнес-логіки без складних перетворень. Основною відмінністю NoSQL систем від реляційних є принцип роботи з неструктурованими та напівструктурованими даними, що не потребують попереднього визначення схеми. Це забезпечує гнучкість і дозволяє швидко реагувати на зміни у форматах даних. На відміну від традиційної моделі ACID, у NoSQL системах часто використовується модель BASE (Basically Available, Soft state, Eventually consistent). Вона орієнтована на забезпечення високої доступності та масштабованості за рахунок ослаблення вимог до миттєвої узгодженості даних. Таким чином, у розподілених середовищах дані можуть тимчасово бути неузгодженими, але з часом система досягає консистентного стану. Серед переваг NoSQL баз даних – висока швидкість обробки запитів, масштабованість, гнучкість структури, простота розподілу даних між кількома серверами. Проте вони мають і недоліки: відсутність єдиного стандарту запитів, обмежена підтримка транзакцій, складність забезпечення цілісності даних та необхідність додаткової логіки на рівні додатку.

Нереляційні бази даних широко застосовуються у сферах, де обробляються великі обсяги динамічної або неструктурованої інформації, наприклад, у соціальних мережах, системах аналітики, e-commerce, потоковій обробці даних та IoT-проєктах. Їхня архітектура робить можливим ефективне зберігання і швидкий доступ до даних у середовищах з постійно зростаючими навантаженнями.

Порівняльний аналіз архітектур SQL та NoSQL баз даних дозволяє глибше зрозуміти суттєві відмінності між цими двома підходами до зберігання та обробки інформації. Основна різниця між ними полягає у структурі організації даних, механізмах забезпечення транзакційності, принципах масштабування, продуктивності та способах підтримки цілісності.

У реляційних базах даних (SQL) структура є строго визначеною: дані організовані у вигляді таблиць, які пов'язані між собою ключами. Кожна таблиця має фіксовану схему, а зміни у структурі вимагають модифікації всієї бази. Такий підхід забезпечує високу передбачуваність і логічну цілісність даних, однак обмежує гнучкість системи. На відміну від цього, у нереляційних базах даних (NoSQL) інформація може зберігатися у вигляді документів, графів або пар "ключ–значення". Така структура не потребує заздалегідь визначеної схеми, що робить її гнучкішою й зручнішою при роботі з динамічними або неструктурованими даними. Завдяки цьому NoSQL системи легше адаптуються до змін у моделі даних без необхідності повної реконфігурації схеми.

Ще однією фундаментальною відмінністю є спосіб забезпечення транзакційності. Реляційні системи дотримуються моделі ACID (Atomicity, Consistency, Isolation, Durability), яка гарантує повну узгодженість і надійність даних.

NoSQL бази натомість використовують більш гнучкий принцип BASE (Basically Available, Soft state, Eventually consistent), який допускає тимчасову неузгодженість даних в обмін на підвищену швидкодію та масштабованість. Таким чином, якщо SQL орієнтується на

стабільність і точність, то NoSQL – на ефективність і доступність у розподілених середовищах.

Важливу роль у порівнянні відіграє також масштабованість систем. Реляційні бази даних зазвичай підтримують вертикальне масштабування, тобто збільшення потужності одного сервера. Це ефективно при середніх навантаженнях, але стає проблематичним при обробці дуже великих обсягів даних. Натомість NoSQL бази підтримують горизонтальне масштабування, яке передбачає розподіл даних між кількома вузлами мережі. Це забезпечує безперервну роботу навіть при високих навантаженнях і робить NoSQL особливо корисними для великих, розподілених систем.

Щодо продуктивності, SQL системи демонструють стабільну роботу при складних транзакційних операціях, де необхідна точність і узгодженість результатів. Проте при виконанні масових операцій запису або обробці великих неструктурованих даних NoSQL рішення зазвичай працюють швидше, оскільки не потребують складних перевірок цілісності на кожному кроці. Крім того, за рахунок розподіленої архітектури вони ефективніше використовують ресурси при обробці запитів у реальному часі.

У питаннях безпеки й цілісності реляційні бази традиційно мають перевагу завдяки розвиненим механізмам контролю доступу, шифрування та журналювання транзакцій. NoSQL системи, хоча й поступово впроваджують подібні засоби, часто покладаються на реалізацію безпеки на рівні додатку або інфраструктури, що може бути як перевагою, так і потенційним ризиком залежно від сценарію використання.

Висновки. У підсумку можна зазначити, що SQL системи забезпечують структурованість, стабільність і надійність, тоді як NoSQL орієнтовані на гнучкість, масштабованість і швидкодію. Вибір між ними залежить від конкретних вимог: якщо потрібна транзакційна узгодженість і чітка схема даних, то варто обрати SQL; якщо ж система має працювати з великими обсягами динамічної інформації, то більш доцільним є використання NoSQL підходу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Redmond, E., & Wilson, J. R. (2018). *Seven databases in seven weeks: A guide to modern databases and the NoSQL movement* (2nd ed.). Pragmatic Bookshelf.

[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, 6*(4), 1–14.

[3] Stonebraker, M., & Çetintemel, U. (2005). “One size fits all”: An idea whose time has come and gone. *Communications of the ACM, 51*(12), 10–15.

Antonenko P. P., Nesterov Yu. M.

Comparative analysis of SQL and NoSQL database architectures in information systems

Abstract. *The paper presents a comparative analysis of relational and non-relational database architectures in the context of modern information systems. Structural differences, integrity models, scalability, performance, and areas of effective application for both approaches are examined. The advantages and disadvantages of each architecture are identified depending on data types and system workload.*

Keywords: *SQL; NoSQL; databases; information systems; architecture.*