

УДК 004.65

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ РІЗНИХ ТИПІВ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ В БАНКІВСЬКІЙ ДІЯЛЬНОСТІ

Козиревич Ярослав Роланович

к.т.н., доцент каф. ІС Глава Марія Геннадіївна

Національний університет «Одеська політехніка», УКРАЇНА

АНОТАЦІЯ. У даній роботі проводиться аналіз різних типів СУБД, включаючи реляційні, об'єктно-орієнтовані та NoSQL, з метою встановлення їх особливостей, можливостей та ефективності в банківській діяльності. Результати дослідження дозволять визначити найбільш ефективні типи СУБД та їх доцільне використання.

Вступ. У сучасному світі банківська діяльність ґрунтується на ефективному управлінні великим обсягом даних, що вимагає використання потужних і надійних систем управління базами даних. Реляційні, об'єктно-орієнтовані та NoSQL СУБД відіграють ключову роль у забезпеченні надійного та швидкого доступу до даних, забезпечуючи високу продуктивність та ефективність роботи банківських систем. Дослідження ефективності використання різних типів СУБД у банківській діяльності є актуальним завданням, яке має на меті визначити оптимальні рішення для забезпечення найкращого функціонування банківських систем. Це дослідження передбачає порівняння реляційних СУБД, об'єктно-орієнтованих СУБД та NoSQL СУБД з точки зору їх ефективності, продуктивності та відповідності вимогам банківської діяльності.

Мета роботи. Метою роботи є розробка методики використання різних типів систем управління базами даних в сфері банківської діяльності для підвищення продуктивності, ефективності, якості обробки даних, швидкодії роботи додатків та зниження витрат на обслуговування та розширення систем управління базами даних.

Основна частина роботи. В банківській діяльності завжди використовуються СУБД, наприклад для зберігання клієнтських даних, логінів, рахунків тощо. Для того, щоб підібрати найбільш доцільну СУБД, потрібен аналіз даних з якими працює система, розуміння кількості даних, що будуть зберігатися. Також потрібно враховувати наскільки часто записи будуть змінюватися, додаватися і зчитуватися. Окрім цього, для швидкодії роботи з даними в СУБД є велика кількість функцій, які їх прискорюють, але також можуть і сповільнити при неналежному застосуванні. У реальних проектах системи використовують декілька СУБД, така практика називається polyglot persistence. Потреба в polyglot persistence виникає через відмінність даних з якими працює система. Для одних даних краще використовувати NoSQL, для інших – реляційну СУБД або об'єктно-реляційну.

Для аналізу було вирішено розглянути найпопулярніші СУБД різних моделей на сьогоднішній день: MySQL – реляційна СУБД; Oracle Database – об'єктно-реляційна СУБД; PostgreSQL – об'єктно-реляційна СУБД; MongoDB – NoSQL СУБД; Redis – NoSQL СУБД.

Порівнюючи СУБД, що працюють з SQL, перша помітна відмінність – те, що MySQL дуже проста в використанні, вона має більш обмежену кількість функцій, простий інтерфейс та структуру. Крім того, серед усіх SQL СУБД вона працює найшвидше для операцій читання завдяки дуже оптимізованому двигуну. Таким чином, якщо таблиця даних в банківській діяльності буде використовуватися переважно для зчитування даних або розробникам потрібно буде якнайшвидше впровадити СУБД до системи, але вони ще не мають досвіду роботи з SQL СУБД, рекомендується використовувати саме MySQL.

Друга помітна відмінність SQL СУБД – необхідність платити за велику кількість функцій в Oracle Database, що доступні в MySQL та PostgreSQL, а також за сервер з базою даних. Тому потрібно заздалегідь визначитися, чи вистачає бюджету проекту для доступу до функцій Oracle.

Для операцій видалення, зміни та додавання даних до таблиці, Oracle Database працює приблизно з такою ж швидкістю, що й PostgreSQL, але Oracle Database гарантує більшу кількість виконаних транзакцій за секунду. Це означає, що якщо навантаження на систему велике, багато

транзакцій виконуються паралельно, в такому випадку Oracle в цьому плані буде працювати краще, так як Oracle цілеспрямовано додали велику кількість модулів, що прискорюють роботу паралельних транзакцій [2,3].

Порівнюючи MongoDB з Redis, можна одразу сказати, що Redis використовується для кешування та швидкого доступу до даних в пам'яті, на відміну від MongoDB. Redis, в цілому, швидше ніж MongoDB, тому що дані зберігаються в оперативній пам'яті, а не на диску. Redis має дуже обмежену функціональність запитів, наприклад, їй не доступні функції агрегації, логічні оператори, групування, операції з'єднання і індексування. Redis не дозволяє горизонтальне масштабування. Якщо додаток повинен відповідати ACID вимогам, варто обирати MongoDB. Наприклад, Redis не гарантує постійність даних, тобто їх довгострокове збереження в базі даних, якщо процес аварійно завершився, всі дані будуть втрачені. Окрім цього, Redis може зберігати досить малу кількість даних, на відміну від MongoDB. В Redis не можна зробити Rollback транзакції, що може призвести до порушення консистентності даних.

З використанням Redis все очевидно, але MongoDB досить часто використовують замість SQL СУБД, тому потрібно зрозуміти, коли варто віддати перевагу саме MongoDB чи SQL СУБД. По-перше, NoSQL СУБД набрали свою популярність відносно нещодавно, тому є ймовірність, що розробникам доведеться вивчати синтаксис та основні функції MongoDB. Друга важлива відмінність – можливість MongoDB горизонтально масштабуватися, тобто можливість розбивати сервер бази даних на декілька серверів для збільшення швидкодії. В нових версіях MongoDB набула повної підтримки ACID вимог, але в SQL СУБД дані вимоги більш оптимізовані, тому, якщо потрібна повна підтримка ACID, слід віддати перевагу саме SQL СУБД. Також, в NoSQL СУБД легко втратити цілісність даних, тому що дані СУБД не використовують зовнішні та первинні ключі, обмеження даних та тригери, в той час як в SQL СУБД накладається купа умов для коректної роботи, що дозволяє досягти цілісності даних. В MongoDB замість таблиць використовуються документи – Json об'єкти, і якщо в SQL СУБД дані було б розподілено на декілька таблиць, в MongoDB весь набір даних буде зберігатися в одному об'єкті, що дозволить робити вибірку даних набагато швидше, оскільки не потрібно зіставляти дані з різних таблиць. Остання відмінність полягає в тому, що в MongoDB нема чіткої структури даних, документи можуть містити абсолютно різні поля, ці поля можна спокійно видаляти або можна додавати будь-які додаткові поля тільки до певних документів, тому якщо структура БД ще невідома або часто змінюється, слід віддати перевагу MongoDB. Виходячи з вищесказаного, можна сказати, що швидкість роботи з записами БД, звичайно, більша у MongoDB ніж в SQL СУБД, завдяки можливості горизонтального масштабування, відсутності обмежень та перевірок на дані, з якими система працює, та завдяки збереженню всіх даних в одному документі.

Роботу будь-якої СУБД можна у разі прискорити за допомогою належного використання так званих некластеризованих індексів. Існує величезна кількість типів індексів, найбільш популярні: В-дерево, та Hash таблиця. Окрім цього, існують такі індекси як: R-дерево, Bitmap, GiST, SP-GiST, GIN та BRIN. Говорячи про найбільш популярний тип індексу – В-дерево, який покриває 90 відсотків випадків, він прискорює операції роботи з СУБД у десятки, а то і сотні разів. Наприклад, якщо звичайний пошук значення по таблиці виконується за алгоритмічною складністю $O(n)$, то В-дерево виконує пошук за $O(\log n)$, тобто, чим більше записів в таблиці, тим помітніша буде різниця. Варто відмітити, що індекси слід використовувати обережно, індекс – це по суті окрема таблиця, яка посилається на основну таблицю, і дані, що змінюються в основній таблиці, тепер потрібно буде змінювати і в таблиці індексу. Тому велика кількість індексів на одну і ту ж таблицю може, навпаки, сповільнити роботу. Індекси слід використовувати на тих атрибутах таблиці, що часто зчитуються та рідко змінюються. Також, якщо кількість записів в таблиці менше приблизно 1000, то створення індексу не дасть значного приросту швидкодії [1].

У таблиці 1 наведено порівняння швидкості зчитування одного запису з середини таблиці СУБД без некластеризованих індексів (1), видалення, зміна та додавання елементу мало приблизно таку ж кореляцію. Порівняння швидкості відбувалося за допомогою інструменту Apache JMeter. Таблиця СУБД не містить зовнішніх ключів, що сповільнило б роботу SQL СУБД

при спробі зчитати дані з дочірньої таблиці, або з батьківської з посиланням на дочірню, на відміну від MongoDB.

Таблиця 1 – Порівняльні характеристики СУБД

| Назва СУБД | Розмір таблиці/документу 100 записів | | Розмір таблиці/документу 1000 записів | | Розмір таблиці/документу 10000 записів | | Розмір таблиці/документу 100000 записів | | Розмір таблиці/документу 1000000 записів | |
|------------|--------------------------------------|---------|---------------------------------------|---------|----------------------------------------|---------|-----------------------------------------|---------|------------------------------------------|-------|
| | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Oracle DB | 7 мс | 7 мс | 30 мс | 9 мс | 97 мс | 11 мс | 853 мс | 14 мс | 4.4 с | 17 мс |
| MySQL | 6 мс | 6 мс | 26 мс | 8 мс | 82 мс | 10 мс | 801 мс | 13 мс | 4 с | 15 мс |
| MongoDB | 5 мс | 5 мс | 23 мс | 7 мс | 70 мс | 8 мс | 720 мс | 12 мс | 3.5 с | 14 мс |
| Redis | 0.2 мкс | 0.2 мкс | 0.2 мкс | 0.2 мкс | 0.2 мкс | 0.2 мкс | 0.4 мкс | 0.4 мкс | 1 мс | 1 мс |
| PostgreSQL | 7 мс | 7 мс | 29 мс | 9 мс | 95 мс | 10 мс | 836 мс | 14 мс | 4.3 с | 17 мс |

Примітка: 1 – без індексу; 2 – з індексом

У таблиці 1 також наведено порівняння швидкості зчитування одного запису з середини таблиці СУБД після створення В-дерево індексу (2). При доцільному використанні індексів швидкість запитів до БД значно прискорюється. Слід відмітити, що Redis не використовує індекси, тому швидкість для Redis залишається однаковою.

Говорячи про сферу банківської діяльності, на основі аналізу різних СУБД рекомендовано використовувати Redis для збереження сесій клієнтів та дані, що найчастіше використовуються користувачами. MongoDB слід використовувати на початкових етапах розробки проекту, коли структура ще не встановлена точно, а також для даних, що можуть мати різну структуру, наприклад, документи, звіти, юридичні угоди, активність користувачів тощо. MySQL підходить для таблиць, які переважно зчитуються та рідко змінюються, наприклад, облікові записи клієнтів. В інших випадках слід віддати перевагу PostgreSQL чи Oracle, з урахування того, що Oracle потребує платну ліцензію. Важливо також пам'ятати про використання індексів, розуміючи їх належне використання. Останнє, що потрібно відмітити, це необхідність шифрувати такі дані, як наприклад, паролі користувачів. Сучасні СУБД мають досить надійний захист, але все одно, слід пам'ятати про безпеку та не зберігати персональні дані у незахищеному виді.

Висновки. У ході даної роботи виконано аналіз найпопулярніших СУБД різних моделей, що використовуються в банківській сфері, запропоновано додаткові функції, що пришвидшують роботу СУБД, згідно з чим було запропоновано методологію застосування розглянутих СУБД. Також проведено порівняння швидкості роботи СУБД з використанням індексів та без них для операцій зчитування, видалення, додавання та зміни.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «14 питань про індекси, які ви соромилися спитати», Стаття. [Online]. Available: <https://habr.com/ru/articles/247373/>. [Accessed: May 12, 2023]
2. «MySQL vs PostgreSQL vs Oracle», Стаття. [Online]. Available: <https://bytescout.com/blog/mysql-vs-postgresql-vs-oracle.html>. [Accessed: May 14, 2023]
3. Arsirii, O. O., Glava, M. G., Kolonko, Matthias & Glumenko, A. O. “Information Technology of Supporting Architectural Solutions Using Polyglot Persistence Concept in Learning Management Systems”. Applied Aspects of Information Technology. Publ. Nauka i Tekhnika. Odessa: Ukraine. 2020; Vol. 3 No.2: 13–31. DOI: <https://doi.org/10.15276/aait.02.2020.1>.

RESEARCH OF THE EFFICIENCY OF USING DIFFERENT TYPES OF DATABASE MANAGEMENT SYSTEMS IN BANKING

Kozyrevych Yaroslav

Ph.D., associate professor, Department IS Glava Maria
Odesa Polytechnic National University, UKRAINE

АНОТАЦІЯ. This paper analyzes various types of DBMS, including relational, object-oriented, and NoSQL, in order to establish their features, capabilities, and effectiveness in banking. The results of the study will allow to determine the most effective types of DBMS and their appropriate use.