УДК 004.658

Д. С. Довгань
М. А. Томчук
П. П. Малініч

СТВОРЕННЯ ЗВ'ЯЗКУ МІЖ НЕРЕЛЯЦІЙНИМИ БАЗАМИ ДАНИХ

Вінницький національний технічний університет;

Анотація

Ця робота досліджує методи створення ефективних зв'язків між нереляційними базами даних, такими як MongoDB. Особливу увагу приділено методологіям інтеграції та синхронізації даних, а також підходам, що забезпечують цілісність і продуктивність інформації в масштабованих системах. Ми обговорюємо основні інструменти, які можуть бути використані для зв'язку між різними базами даних, і розглядаємо практичні застосування в різних контекстах.

Ключові слова: MongoDB, нереляційні бази даних, інтеграція, масштабованість, продуктивність, синхронізація даних, цілісність інформації.

Abstract

This paper explores methods for creating efficient connections between non-relational databases such as MongoDB. Special attention is paid to the methodologies of data integration and synchronization, as well as approaches that ensure the integrity and productivity of information in scalable systems. We discuss the main tools that can be used to communicate between different databases and consider practical applications in different contexts.

Keywords: MongoDB, non-relational databases, integration, scalability, performance, data synchronization, information integrity.

Вступ

З розвитком сучасних технологій та зростанням кількості даних, все більшої популярності набувають нереляційні бази даних, як-от MongoDB. Вони надають переваги у швидкості, гнучкості та масштабованості. Проте, їх використання створює нові виклики, особливо коли необхідно інтегрувати дані між різними системами або забезпечити узгодженість інформації [1]. У цьому дослідженні ми вивчаємо різні підходи та технології, які дозволяють створити зв'язки між нереляційними базами даних, зокрема, коли потрібна висока продуктивність та масштабованість.

Огляд можливостей

Для створення зв'язків між нереляційними базами даних, такими як MongoDB, існує кілька ключових підходів, кожен із яких має свої переваги та недоліки. У цьому розділі ми розглянемо ці підходи, а також опишемо власний досвід використання ідентифікаторів для встановлення зв'язків між даними.

Одним із найпоширеніших підходів для інтеграції даних між нереляційними базами даних є використання API-інтерфейсів або веб-сервісів. Цей підхід дозволяє різним системам взаємодіяти через стандартизовані інтерфейси, що забезпечує гнучкість і можливість інтеграції в реальному часі. API-інтерфейси забезпечують високу гнучкість, дозволяючи різним додаткам і сервісам спілкуватися через мережу. Вони також можуть використовуватися для під'єднання до зовнішніх сервісів або інших баз даних, що робить цей підхід масштабованим та легким у застосуванні [2]. Основним недоліком є необхідність підтримки та оновлення API-інтерфейсів, а також можливість виникнення мережевих затримок або збоїв у зв'язку. Також, взаємодія через API може потребувати додаткової аутентифікації та безпеки [3].

Канали повідомлень є ще одним підходом для передачі даних між системами. Вони забезпечують асинхронну передачу повідомлень, що дозволяє системам взаємодіяти без прямої залежності від мережевого зв'язку. Канали повідомлень забезпечують високу відмовостійкість, оскільки системи можуть працювати незалежно від інших. Вони також дозволяють масштабувати передачу даних, оскільки системи можуть обробляти повідомлення в будь-який зручний час. Можлива затримка в

обміні даними, оскільки повідомлення можуть оброблятися асинхронно. Крім того, налаштування та обслуговування каналів повідомлень може бути складним завданням.

Реплікація даних – це процес створення копій даних у різних базах даних або системах, щоб забезпечити цілісність і доступність інформації. Резервне копіювання також допомагає зберігати дані в безпечному місці для запобігання втратам. Реплікація забезпечує високий рівень доступності даних та стійкість до збоїв. Вона також допомагає зберегти цілісність даних у випадках аварійних ситуацій. Однак реплікація може бути ресурсомісткою та вимагати додаткових обчислювальних потужностей. Також існує ризик виникнення конфліктів у разі одночасного оновлення даних [4].

Унікальні ідентифікатори використовуються для створення зв'язків між різними частинами баз даних. Цей підхід дозволяє побудувати логічну структуру даних та забезпечити їхню цілісність. Кожен запис у базі даних отримує унікальний ідентифікатор, який використовується як ключ для зв'язку з іншими записами або об'єктами. Така структура дозволяє встановлювати зв'язки між різними рівнями даних або різними таблицями в базі даних.

Переваги підходу:

- Взаємозв'язок: Унікальні ідентифікатори забезпечують легку інтеграцію різних рівнів даних, дозволяючи будувати складні ієрархії та зв'язки.
- Цілісність даних: Використання унікальних ідентифікаторів зменшує ймовірність дублювання записів або плутанини в даних, оскільки кожен об'єкт може бути пов'язаний лише з одним унікальним ідентифікатором.
- Зручність масштабування: Такий підхід полегшує додавання нових даних і розширення структури бази даних, оскільки зв'язки чітко визначені.

Недоліки підходу:

- Складність структури: Зі збільшенням складності бази даних підтримка унікальних ідентифікаторів може стати складнішою, особливо коли йдеться про збереження зв'язків у великих системах.
- Проблеми синхронізації: Якщо дані реплікуються або синхронізуються між різними базами даних, можуть виникати складнощі з підтриманням цілісності унікальних ідентифікаторів.
- Підтримка цілісності: У випадку видалення чи оновлення записів необхідно забезпечити, щоб зв'язки зберігалися або коректно оновлювалися, що може вимагати додаткової уваги.

Використання унікальних ідентифікаторів — це ефективний спосіб забезпечити структурування та цілісність баз даних. Це допомагає не лише створити зв'язки між різними елементами, але й полегшує інтеграцію даних між різними системами або додатками. Проте важливо пам'ятати про потенційні складнощі з підтримкою та масштабуванням, особливо в великих системах.

Висновки

Існують різні підходи до встановлення зв'язків між нереляційними базами даних, кожен із яких має свої переваги та недоліки. Використання API-інтерфейсів і каналів повідомлень забезпечує високу гнучкість та масштабованість, тоді як реплікація та резервне копіювання допомагають зберегти цілісність даних. Унікальні ідентифікатори у MongoDB можуть стати ефективним способом зв'язку між різними рівнями даних, дозволяючи зберігати логічну структуру баз даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Паламарчук Є. А. Бази даних та інформаційні системи [Текст] : навчальний посібник / Є. А. Паламарчук – Вінниця: ВНТУ, 2017. – 86 с.

2. Пєтух А. М. Бази даних. Мови запитів, управління транзакціями, розподілена обробка даних [Текст] : навчальний посібник / А. М. Пєтух, О. В. Романюк, О. Н. Романюк. – ВНТУ, 2016. – 95 с.

3. Чернишов К. А. Методи збору даних досвіду взаємодії користувача для випробувального етапу розробки через тестування [Текст] / К. А. Чернишов, І. П. Малініч, П. П. Малініч // Міжнародна наукова інтернет-конференція "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення, м. Тернопіль, 5 лютого 2019 р. : збірник тез доповідей. – Тернопіль, 2019. – Вип. 35. – 43 с.

4. Indexes – MongoDB Manual [Електроний ресурс] – Режим доступу: https://www.mongodb.com/docs/manual/indexes/

Довгань Дмитро Сергійович – студент групи 1КІ-20б, факультет Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця

Томчук Микола Антонович – канд. техн. наук, доцент кафедри Обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: tomchuk@vntu.edu.ua

Малініч Павло Павлович – асистент кафедри Програмного забезпечення, Вінницький національний технічний університет

Dmytro Dovhan – Student in Group 1KI-20b, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia

Mykola Tomchuk – Cand. Sc. (Tech), Docent of Computer Technologies department, Vinnytsia National Technical University, Vinnytsia, email: tomchuk@vntu.edu.ua

Pavlo Malinich – Assistant Lecturer of Software Development department, Vinnytsia National Technical University, Vinnytsia