НОВИЙ ПІДХІД ДО ЗБЕРЕЖЕННЯ СЛАБОСТРУКТУРОВАНИХ МЕДИЧНИХ ДАНИХ

І.Б. Швороб¹

У зв'язку зі швидким збільшенням обсягу слабоструктурованих і неструктурованих даних, питання про їх оптимальне збереження є досить актуальним. Важливо зберігати їх в зручному форматі для подальшого оброблення. Проте обробити неструктуровані або слабоструктуровані дані складно через невизначеність у схемі даних. Щоб усунути цю проблему, здійснено аналіз нереляційних баз даних та їх застосування на реальному прикладі для збереження системи і оброблення медичних даних.

Ключові слова: слабоструктуровані дані, NoSQL, Neo4j, графові бази даних, документо-орієнтовані бази даних.

Вступ. Швидке збільшення обсягу інформації зумовило пошук нових підходів до вирішення проблеми з її збереження. Вважають, що нереляційні бази даних (NoSQL) найбільш придатні для збереження слабоструктурованих даних. NoSQL охоплює широкий спектр технологій баз даних, розроблених згідно з вимогами нових систем та їх потреб [1, 11]:

- розробники працюють у середовищі, що продукує велику кількість нових, швидкозмінних типів даних – структурованих, частково структурованих, неструктурованих і поліморфних даних;
- минув час, коли використовували водоспадну (каскадну) модель розроблення програмного забезпечення. Це потребувало досить багато часу. На цей час використовують гнучкі моделі для розроблення програмного забезпечення з чітким поділом на спринти, ітерації. Завдяки цьому досягають запланованих змін у коді щотижня або двічі на тиждень, а інколи навіть по кілька разів на день;
- програми для користувацької аудиторії на цей час представляються в якості послуг, які повинні бути завжди доступними незалежно від пристрою чи системи і масштабуватись по всьому світу для мільйонів користувачів;
- організації частіше використовують масштабовані архітектури з використанням вільного програмного забезпечення, розподілених серверів і хмарних обчислень замість великих монолітних серверів та інфраструктури зберігання даних.

Реляційні бази даних не були розроблені для того, щоб впоратися із проблемами, які виникли сьогодні, не були побудовані для того, щоб забезпечити паралелізм даних, скористатися перевагами сховищ і обчислювальної потужності, доступних сьогодні.

Нереляційні бази даних діляться на кілька типів залежно від можливості масштабування систем зберігання, моделі даних і запитів. Основні види нереляційних баз даних: бази даних типу ключ/значення, документо-орієнтовані, стовпчиково-орієнтовані та графові бази даних. Бази даних типу ключ/значення використовують для організації даних. Це дає змогу зберегти за певним ключем будь-які дані. У документарно-орієнтованих базах даних кожен запис зберігається у вигляді окремого документа, який має свій власний набір полів. Стовпчиково-орієнтовані бази даних зберігають дані не як кортежі, а як стовпці. Для подання баз даних у вигляді графів, використовують вершини і ребра, які з'єд-

нують їх [2]. Вибір типу нереляційної бази даних залежить від багатьох факторів, зокрема й обраної предметної області.

Порівняно з реляційними базами даних, бази даних NoSQL гнучкіші у плані масштабування та забезпечують кращу продуктивність, а також з допомогою їх моделей даних вдається вирішити кілька проблем, нетипових для класичної реляційної моделі:

- великі обсяги швидко змінюваних структурованих, слабоструктурованих і неструктурованих даних;
- чіткі часові межі для опрацювання певного обсягу даних, швидка ітерація схеми, а також часті зміни коду;
- простота та зручність у використанні для об'єктно-орієнтованого програмування;
- географічно розподілена масштабована архітектура, замість дорогої, монолітної архітектури.

Мета роботи – порівняння та аналіз графових і документо-орієнтованих баз даних, наведення прикладів опрацювання та збереження даних для відповідних БД для медичних даних та об'єднання обох видів баз даних.

1. Отримання інформації з деяких баз даних NoSQL

1.1. Документо-орієнтовані бази даних

Документо-орієнтовані бази даних характеризуються тим, що в них немає схеми організації даних [3]. Це означає:

- сутності не повинні мати однорідну структуру, тобто різні записи можуть мати різні стовпці;
- типи значень можуть бути різними для кожного запису;
- колонки можуть мати більше одного значення (масиви);
- записи можуть мати вкладену структуру.

Документо-орієнтовані БД часто використовують внутрішні позначення, які можуть бути опрацьовані безпосередньо в додатках, в основному JSON. JSON-документи зазвичай також можуть бути збережені у вигляді чистого тексту в ключ/значення-сховищах або реляційних системах управління базами даних. Проте такі документи вимагають оброблення структур на стороні клієнта, а це, водночас, робить недоступними деякі можливості, що надаються сховищами документів (наприклад, вторинні індекси).

Документи подаються у вигляді одно- або дворівневих конструкцій типу ключ-значення. Можна сформувати документ будь-якої складної структури із збереженням можливості їх використання у заданій системі управління. Незважаючи на потужний характер і можливість отримати записи окремих ключів, системи управління на основі документів мають свої власні проблеми порівняно з іншими. Наприклад, отримання значення запису означає отримання всієї інформації про запис. Те ж саме стосується оновлень, а це впливає на продуктивність. Дані і відношення не зберігаються в таблицях, є схожими на звичайні реляційні бази даних, але насправді представляють собою набір незалежних документів.

Документо-орієнтовані бази даних часто використовують для:

¹ аспірт. І.Б. Швороб – кафедра ІСМ, НУ "Львівська політехніка"

- *ущільненої інформації* – документи на основі сховища даних дають змогу працювати з глибоко вкладеними, складними структурами даних.
- *JavaScript-дружніх систем* – одним з найважливіших функціональних документів на основі сховища даних є спосіб, яким вони взаємодіють з додатками: за допомогою JS-дружнього подання даних JSON.

Найвідомішими документо-орієнтованими базами даних є:

- CouchDB – документо-орієнтована система управління базами даних, яка не потребує опису схеми даних. Ця програма є вільною, відкритою, написана на мові Erlang [5].
- MongoDB – документо-орієнтована система управління базами даних з відкритим вихідним кодом, яка не потребує опису схеми таблиць. Написана на мові C++. СУБД оперує наборами JSON-подібних документів, що зберігаються у двійковому вигляді у форматі BSON.

Є шість основних концепцій для документо-орієнтованих баз даних на прикладі MongoDB [4]:

1. MongoDB – концептуально те ж саме, що і звичайна реляційна база. Всередині MongoDB може бути нуль або кілька баз даних, кожна з яких є контейнером для інших об'єктів;
2. База даних може мати нуль або більше "колекцій" (рис. 1). Колекція настільки схожа на традиційну "таблицю", що її можна розглядати, як те ж саме;
3. Колекція складається з нуля або більше "документів". Знову ж таки, цей документ можна розглядати як "рядок";
4. Документ складається з одного або кількох "полів", які можна розглядати як "стовпці";
5. "Індекси" в MongoDB практично ідентичні індексам у реляційних базах даних;
6. "Курсори" відрізняються від попередніх п'яти понять, але вони дуже важливі (хоча іноді це ігнорується) і заслуговують окремого обговорення. Важливо розуміти, що коли здійснюється запит будь-яких даних в MongoDB, повертається покажчик, за допомогою якого можна зробити що-небудь – розрахувати, пропустити певну кількість попередніх записів – без завантаження даних.

Схематичне подання документо-орієнтованої бази даних подано на

рис. 1.

Рис. 1. Схематичне подання документо-орієнтованої бази даних

Прикладом такого документа на основі обраної предметної області (несструктуровані файли із вмістом інструкцій до медичних ліків) у форматі JSON є:

```

{
  "title": "Levorin",
  "latin_title": "Levorinum",
  "indication": [
    "Candidiasis gastrointestinal tract",
    "Lesion of skin",
    "Mucosal lesions"],
  "contraindication": [
    "Liver",
    "Acute gastrointestinal diseases",
    "Gastric ulcer and duodenal ulcer",
    "Uterine bleeding"],
  "release_form": [
    "Powder for external use",
    "Tablets 500 000 IU",
    "Tablets 500 000 IU",
    "2000000 IU powder",
    "cream in tubes of 30 meters"],
  "application_method": [
    "Inside of 400,000-500,000 IU 2-3 times a day for 10-12 days",
    "Children under 2 years are designated on the basis of 25 000 IU / kg per day",
    "From 2 to 6 years – 20 000 IU / kg per day",
    "After 6 years – in a single dose of 200,000-250,000 IU 3-4 times a day"]
}

```

де: title – назва препарату; latin_title – загальна назва латиною; indication – показання для застосування; contraindication – протипоказання; release_form – форма випуску лікарського засобу; application_method – дозування.

1.2. База даних на основі графів

Графові бази даних, також відомі як графо-орієнтовані бази даних, – графи, подання даних в яких здійснюється у вигляді структур, таких як вузли, та ребер, які є відношеннями між вузлами [6]. Такі бази даних дають змогу легко обробляти дані в цій формі і легко обчислити специфічні властивості графів, таких як довжина шляху від одного вузла до іншого.

Графо-орієнтовані бази даних, як правило, не забезпечують індекси для всіх вершин, прямий доступ до вузлів на основі значень атрибутів не є можливим у цих випадках. Моделі на основі графових баз даних використовують деревовидні структури (тобто графи) з вузлами і ребра, що з'єднують між собою через відношення (рис. 2).

Деякі операції над даними набагато простіше виконати з використанням цього типу моделей, завдяки своїм зв'язкам і групам пов'язаних елементів інформації (наприклад, пов'язаних людей). Ці бази даних зазвичай використовуються додатками, при цьому чіткі межі для підключень потрібно встановити. Наприклад, під час реєстрації в соціальній мережі будь-якого роду набагато простіше працювати з використанням систем управління базами даних, що базуються на графах.

Графові бази даних, насамперед, призначені для вирішення тих задач, де дані тісно зв'язані між собою у відношеннях, які можуть заглиблюватись в кілька рівнів. Графові бази даних призначені для вирішення проблем, коли дані можуть бути віддалені один від одного на два або більше зв'язки. Це вирішується дуже просто, коли дані моделюються як "вершини графів", а зв'язки як "ребра графів" між цимим вузлами. Також перевагою є те, що можна здійснювати обхід графа з допомогою давно відомих та ефективних алгоритмів.

Найчастіше графові бази даних використовують для:

- опрацювання складної реляційної інформації – бази даних на основі графів роблять цей процес надзвичайно ефективним і простим у використанні зі складними структурами даних, але не мають реляційної інформації, наприклад, зв'язків між двома об'єктами і різними ступенями інших суб'єктів, опосередковано пов'язаних з ними.
- моделювання та класифікація – графо-орієнтовані бази даних стануть найкращим рішенням у будь-якій ситуації, де беруть участь відношення. Моделювання даних і класифікація різної інформації, що зв'язана реляційним чином, можуть бути оброблені дуже добре за допомогою цього типу сховищ даних.

Найвідоміші графові бази даних:

- Neo4J пропонує повноцінну базу даних із транзакціями, індексами, кількома режимами роботи і простотою вивчення, завдяки дуже легкій структурі [7]. Весь код відкритий і доступний, і програма розповсюджується під ліцензією AGPL для некомерційного використання, а також є платні варіанти для комерційного.
- AllegroGraph є сучасною, високопродуктивною, постійною графовою базою даних [8]. AllegroGraph використовує ефективне використання пам'яті в поєднанні зі зберіганням на основі жорстких дисків, що дає змогу йому масштабувати до мільярдів квадрантів, зберігаючи при цьому високу продуктивність. AllegroGraph підтримує SPARQL, RDFS++ і логіку Prolog для великого числа клієнтських програм.

Рис. 2. Схематичне подання графової бази даних

2. Документ на основі бази даних графа

Під час роботи зі слабоструктурованими даними важливо зберегти якомога більшу їх кількість у найшвидшій для використання формі. База даних на основі документо-орієнтованого графа включає складність вузла графа даних, тобто, коли вузлом є елемент з багатьма різними характеристиками.

Прикладами такого роду баз даних є Orientdb, CouchDB, Neo4j [10]. Така реалізація використовує документ для забезпечення гнучкості запитів до гра-

фових баз даних, а збереження у вигляді ключ/значення забезпечує швидкий пошук даних. Об'єкт G такої бази даних можна подати так:

$$G = \{ \langle N, E \rangle \}, \quad (1)$$

де: N – вершина графа, E – множина ребер,

$$E = \{ \langle e_1, \dots, e_n \rangle \}. \quad (2)$$

З огляду на те, що документ використовує спрямований граф, то вузол графа можна подати у вигляді об'єкта, який містить множину параметрів типу ключ/значення, наприклад,

$$N = \{ \langle k, v \rangle \}. \quad (3)$$

Отже, документо-орієнтований граф можна подати у такому вигляді:

$$G = \{ \langle \{k, v\}, \{e_1, \dots, e_n\} \rangle \}. \quad (4)$$

На рис. 3 показано схематичне подання документо-орієнтованого графа.

Рис. 3. Схематичне подання документо-орієнтованого графа

Розглянемо приклад збереження та оброблення даних про лікарські засоби (рис. 4), побудувавши документо-орієнтований граф на основі бази даних Neo4j [9].

Створені вершини графа будуть двох типів: препарати і хвороби (показання та протипоказання). Ребра також будуть двох типів: показання та протипоказання. Наступний приклад показує створення вузла і ребер для медичних препаратів. Оскільки це документо-орієнтований граф, то під час створення вершин вказуємо додаткові характеристики.

```
CREATE (Levorinum:Antibiotic {title:'Levorin', latin_title:'Levorinum', release_form: 'Tablets 500 000 IU',
application_method:'Inside of 400,000-500,000 IU 2-3 times a day for 10-12 days'})
CREATE (Candidiasis:Disease {name: Candidiasis gastrointestinal tract'})
CREATE (SkinLesions:Disease {name: Lesion of skin'})
CREATE (MucosalLesions:Disease {name: Mucosal lesions'})
CREATE (Liver:Disease {name: Liver illness'})
CREATE (Stomach:Disease {name: Acute gastrointestinal diseases'})
CREATE (Ulcer:Disease {name: Gastric ulcer and duodenal ulcer'})
CREATE (UterineBleeding:Disease {name: Uterine bleeding'})
CREATE
(Candidiasis)-[:INDICATION]->(Levorinum),
(SkinLesions)-[:INDICATION]->(Levorinum),
(MucosalLesions)-[:INDICATION]->(Levorinum),
(Liver)-[:CONTRAINDICATION]->(Levorinum),
```


(Stomach)-[:CONTRAINDICATION]->(Levorinum),
(Ulcer)-[:CONTRAINDICATION]->(Levorinum),
(UterineBleeding)-[:CONTRAINDICATION]->(Levorinum)

Внаслідок цього отримаємо граф з десяти препаратів (рис. 5).

Рис. 4. Документо-орієнтований граф для системи збереження та оброблення інформації про медичні препарати

Запити в цій базі даних будуть виглядати таким чином. Припустимо, що потрібно знайти ліки при показанні до застосування у вигляді деякої інфекції (див. рис. 5):

MATCH (p:Disease)-[r:INDICATION]->(m:Antibiotic) WHERE p.name = "Infection" RETURN p, r, m

Рис. 5. Результат виконання запиту при заданні певної умови

При виборі препаратів потрібно враховувати не тільки показання, але і протипоказання. Нехай протипоказані гострі шлунково-кишкові захворювання. Такий запит буде виглядати так:

MATCH (p:Disease)-[:INDICATION]->(m:Antibiotic)<-[:CONTRAINDICATION]-(pc:Disease)
WHERE p.name = "Infection" AND NOT pc.name = "Acute gastrointestinal diseases"
RETURN m.latin_title

Внаслідок цього отримаємо множину з трьох медичних препаратів: Lactamoxef, Tetracyclin, Lincomycin. Отже, база даних на основі документо-орієнтованого графа в цьому випадку є дуже зручним засобом для збереження даних. Вважається, що при виборі препаратів потрібно враховувати не тільки показання та протипоказання, а також інші фактори, такі як дозування, вік і вага пацієнта та ін. Подальші дослідження будуть спрямовані на розширення та оптимізацію бази даних для врахування цих факторів.

Було здійснено аналіз оброблення слабоструктурованих даних для різних типів баз даних (табл.). Аналіз проводився за такими параметрами: кількість створюваних об'єктів (документів або вузлів) (N), вага бази даних (W), час запису в базу даних (t), час виконання запиту з кількома умовами (t_c). Під час аналізу було використано 100 інструкцій для медичних препаратів.

Табл. Результати аналізу оброблення слабоструктурованих даних

	N	W (Мб)	t (мс)	t_c (мс)
Документо-орієнтована БД	100	40.2	10	2
Графова БД	740	30.9	15	1.4
Документо-орієнтований граф	740	61.1	20.72	1.3

Висновок

Розглянуті бази даних не є однотипними, але в більшості випадків застосовуються в одних і тих же цілях. Хоча для цього використовуються різні засоби. Варто також звернути увагу, що БД, які так чи інакше використовують принцип ключ/значення, як правило, дають об'єктно-орієнтований інтерфейс для програміста.

Вибір між технологіями NoSQL залежить від багатьох факторів (постановка завдання, кваліфікація розробника, особливості вимог до обладнання, швидкодія і т. ін.), тому однозначної рекомендації щодо того, яка база даних повинна бути використана, не може бути надано. Проте вважається, що розглянуті типи баз даних використовують нову і достатньо прогресивну технологію, яка за багатьма показниками випереджає перевірені часом реляційні бази даних (MS SQL, MySQL, Oracle тощо), які вже десятки років займають передові позиції на ринку БД. Зважаючи на лавиноподібне зростання користувачів Інтернету та у зв'язку зі збільшенням навантаження на сховища даних, варто, можливо, переглянути класичні підходи до реалізації баз даних та звернути увагу на нові технології, які пропонує ІТ-спільнота.

Істотною перевагою зберігання даних в документо-орієнтованій базі даних є зручність для подальшого оброблення даних. Однак запити будуть складними і при запиті може прийти набагато більше інформації, ніж це потрібно. Це, водночас, впливає на продуктивність. Дані в графовій базі даних займають великий обсяг. Час оптимізації бази даних графа є більшим. Графові бази даних значно домінують над реляційними у разі пошуку в реальному часі з великими обсягами даних. Отже, графові бази даних потрібно використовувати за наявності великих обсягів даних і ресурсів, за умови, що швидке виконання пошуку є дуже важливим.

Залежно від вимог до проекту та для оптимізації і пришвидшення роботи із великими обсягами даних потрібно використовувати відповідні бази даних. Наприклад, для програмного рішення дуже важливим є швидкий пошук, тому для цього можна використати графову базу даних. В окремих випадках можна комбінувати представлення даних у вигляді документо-орієнтованих графових баз даних.

Література

1. NoSQL Database Couchbase. [Electronic resource]. – Mode of access <http://www.couchbase.com/nosql-resources/what-is-no-sql>
2. Ljalyk O. NOSQL Storage Systems: Comparative Analysis and the Prospects for Their Usage in Educational Portals / O. Ljalyk, V. Mandzjuk // Scientific notes of Ternopil National Pedagogical University. Pedagogy. – Ternopil. – 2011. – Vol. 1. – Pp. 234-241.
3. Banker K. MongoDB in action / K. Banker. – Manning, NY, 2012. – Pp. 288.
4. The MongoDB 3.2 Manual. Technical documentation. – 2016. [Electronic resource]. – Mode of access <http://docs.mongodb.com/manual/>
5. CouchDB Technical documentation – 2016. [Electronic resource]. – Mode of access <http://docs.couchdb.org/en/1.6.1/intro/why.html>
6. Renzo A. Survey of Graph Database Models / A. Renzo, C. Gutierrez // ACM Computing Surveys. – 2008. – Vol. 40, No. 1. – Pp. 123-129.
7. Glibovets, A.M. Comparison Neo4 and relational database MySQL. PROCEEDINGS / A.M. Glibovets, A.O. Dobriansky // Computer Science. – 2015. – Vol. 177. – Pp. 108-112.
8. Franz, Incorporated: AllegroGraph. Technical documentation (2016). [Electronic resource]. – Mode of access <http://franz.com/agraph/support/documentation/current/agraph-introduction.html>
9. Robinson I. Graph Databases / I. Robinson, J. Webber, E. Eifrem // O'Reilly Media, Inc. – 2015. – Pp. 25-53.
10. Buerli M. The Current State of Graph Databases / M. Buerli // Cal Poly San Luis Obispo, 2012. – Pp. 123-129.
11. Planet Cassandra. [Electronic resource]. – Mode of access <http://www.planetcassandra.org/what-is-nosql/>

Надійшла до редакції 26.06.2016 р.

Швороб И.Б. Новый подход к сохранению слабоструктурированных медицинских данных

В связи с быстрым увеличением объема слабоструктурированных и неструктурированных данных, вопрос об их оптимальном сохранении является весьма актуальным. Важно сохранять их в удобном формате для дальнейшей обработки. Однако обработать неструктурированные или слабоструктурированные данные сложно из-за неопределенности в схеме данных. Чтобы устранить эту проблему, осуществлен анализ нереляционных баз данных и их применение на реальном примере для сохранения системы и обработки медицинских данных.

Ключевые слова: слабоструктурированные данные, NoSQL, Neo4 j, графовые базы данных, документо-ориентированные базы данных.

Shvorob I.B. New approach to maintaining health semistructured data

In connection with the rapid increase in the volume semistructured and unstructured data, the question of optimal saving is quite important. In optimal storage, it is important saving them in a convenient format for further processing. However, the processing of unstructured and semistructured data is difficult to implement because of the uncertainty in the data schema. To resolve this issue done some review and analysis of non-relational databases and their applications are on a real example for system preservation and processing of information about medicines.

Keywords: semistructured data, NoSQL, Neo4 j, Graph database, document-oriented database.

5. ОСВІТЯНСЬКІ ПРОБЛЕМИ ВИЩОЇ ШКОЛИ

УДК 004.9

ФОРМАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ У ЗАДАЧАХ УПРАВЛІННЯ ВИЩИМИ НАВЧАЛЬНИМИ ЗАКЛАДАМИ ОСВІТИ

В.Є. Мухін¹, Я.І. Корнага², Є.О. Бойко³

Розроблено механізми управління вищим начальним закладом за допомогою засобів підтримки бізнес-процесів. Ефективність управління передбачає застосування методів, які дають змогу проводити заходи щодо забезпечення виконання працівниками навчальної, наукової та методичної роботи. Розроблено механізми підтримки автоматизації бізнес-процесів управлінської діяльності університетів і проведено їх формалізацію та моделювання за допомогою імітаційного моделювання. Проведено моделювання бізнес-процесу за допомогою мережі Петрі, що дало змогу виконати прогнозування появи критичних, недосяжних або тупикових ситуацій та випадків взаємного блокування операцій бізнес-процесів, які можуть призвести до зниження швидкості прийняття рішень у функціонуванні вищих навчальних закладів та їх підрозділів.

Ключові слова: бізнес-процес, формалізація, мережа Петрі.

Вступ. Для функціонування будь-якої організації чи підприємства потрібна автоматизація промислових процесів, яка поєднує в собі сукупність адміністративних та технологічних процесів [1-5]. У закладах освіти важливим є організація управління підрозділами для надання послуг з навчальної, наукової та дослідницької діяльності. Для цього потрібен механізм, який дасть змогу отримувати результат незалежно від багатьох факторів [3-5].

Постановка задачі дослідження. Ефективне управління вищим навчальним закладом передбачає застосування методів, які дають змогу проводити ефективну управлінську діяльність. Виникає задача розроблення механізмів для управління вищим начальним закладом (ВНЗ) за допомогою засобів підтримки бізнес-процесів [6-8].

Структура вищого навчального закладу. У вищому навчальному закладі, як правило, використовують трьохрівневу структуру управління (рис. 1), а саме:

1. Ректорат на чолі з ректором закладу.
2. Факультети (інститути) на чолі з деканом (директором) або департаменти на чолі з керівником департаменту.
3. Кафедри на чолі з завідувачем кафедри або відділи на чолі з начальником відділу.

Ця структура поділяє засоби підтримки бізнес-процесів на три рівні управління:

¹ проф. В.Є. Мухін, д-р техн. наук – НТУ України "Київський політехнічний інститут";
² доц. Я.І. Корнага, канд. техн. наук – НТУ України "Київський політехнічний інститут";
³ аспір. Є.О. Бойко – НТУ України "Київський політехнічний інститут"