

Лабораторна робота №2: Використання NoSQL баз даних

Тема: Система управління проектами в корпорації

1. Види NoSQL баз даних

NoSQL бази даних використовуються для вирішення задач, де реляційні БД мають обмеження (швидкість запису, гнучкість схеми, масштабування). Основні типи:

1. Документоорієнтовані (Document Stores)

- *Приклад:* MongoDB.
- *Структура:* Дані зберігаються як JSON/BSON документи.
- *Застосування:* Зберігання профілів, каталогів, логів, CMS.

2. Ключ-значення (Key-Value Stores)

- *Приклад:* Redis, Memcached.
- *Структура:* Простий масив, де доступ здійснюється по унікальному ключу.
- *Застосування:* Кешування, сесії користувачів, кошики, лічильники.

3. Колонкові (Wide-Column Stores)

- *Приклад:* Cassandra.
- *Структура:* Дані групуються в сімейства колонок.
- *Застосування:* Big Data, IoT, аналітика часових рядів.

4. Графові (Graph Databases)

- *Приклад:* Neo4j.
- *Структура:* Вузли та зв'язки (ребра).
- *Застосування:* Соціальні мережі, побудова маршрутів, рекомендаційні системи.

2. Обґрунтування для "Системи управління проектами"

У нашій системі є модуль "Аудит та Логування дій" (Project Activity Log). Це критичний компонент, який фіксує кожну зміну: створення завдання, зміну статусу, коментар, завантаження файлу.

Чому реляційна БД (SQL) тут неефективна:

1. **Поліморфізм даних:** Подія "Зміна статусу" має поля `old_status` , `new_status` . Подія "Завантаження файлу" має `file_size` , `url` . В SQL доведеться створювати купу колонок з `NULL` або зберігати JSON як рядок.
2. **Write-Heavy навантаження:** У великій корпорації кожна секунда генерує сотні подій. SQL з його ACID транзакціями та блокуваннями може стати вузьким місцем.
3. **Архівування:** Логи швидко ростуть. MongoDB дозволяє легко шардувати (розподіляти) дані.

Рішення: Використати **MongoDB** для збереження логів активності.

3. Проектування структури даних

Варіант SQL (для порівняння)

Ми змушені "сплющувати" структуру або використовувати текстове поле для деталей.

```
CREATE TABLE project_logs_benchmark (  
  log_id INT AUTO_INCREMENT PRIMARY KEY,  
  project_id INT,  
  action_type VARCHAR(50),  
  details_json TEXT, -- Неструктуровані дані тут  
  created_at DATETIME  
);
```

Варіант NoSQL (MongoDB)

Ми зберігаємо об'єкт як є. Схема гнучка.

Приклад документа (Зміна статусу):

```
{  
  "_id": ObjectId("..."),  
  "project_id": 10,  
  "action": "StatusChange",  
  "created_at": ISODate("2023-10-27T10:00:00Z"),  
  "details": {  
    "task_id": 55,  
    "old": "To Do",  
    "new": "In Progress"  
  }  
}
```

Приклад документа (Завантаження файлу):

```
{  
  "_id": ObjectId("..."),  
  "project_id": 10,  
  "action": "FileUpload",  
  "created_at": ISODate("2023-10-27T10:05:00Z"),  
  "details": {  
    "filename": "spec.pdf",  
    "size_mb": 2.5,  
    "path": "/storage/projects/10/"  
  }  
}
```

4. Використання Key-Value (Redis)

Для підвищення швидкодії ми використаємо **Redis** для кешування статистики проекту ("Project Dashboard"). Замість того, щоб щоразу рахувати кількість завдань, бюджет та години через важкі SQL запити, ми будемо зберігати готовий JSON у Redis з часом життя (TTL) 60 секунд.