

Бази даних

Лекція 15

Тематика лекції

- Переваги і недоліки реляційних СУБД
- Типи нереляційних СУБД
- Вибір типу СУБД під задачу

Сильні сторони реляційних субд

Сильні сторони реляційних СУБД

- ACID
- Строга консистентність
- SQL

Слабкі сторони реляційних субд

Слабкі сторони реляційних субд

- Складні запити на великих об'ємах даних
- Висока швидкість і кількість записів
- Надвисокі вимоги до швидкодії
- Ієрархічні структури даних

CAP теорема

C - Consistency - консистентність даних

A - Availability - доступність системи

P - Partition Tolerance - робота при мережевих збоях

Завжди можливий вибір лише двох властивостей з трьох.

CAP теорема

C + P - при порушеннях мережі частина даних стає недоступною.

C + A - при порушеннях мережі система не працює.

A + P - при порушеннях мережі можна отримати застарілі дані.

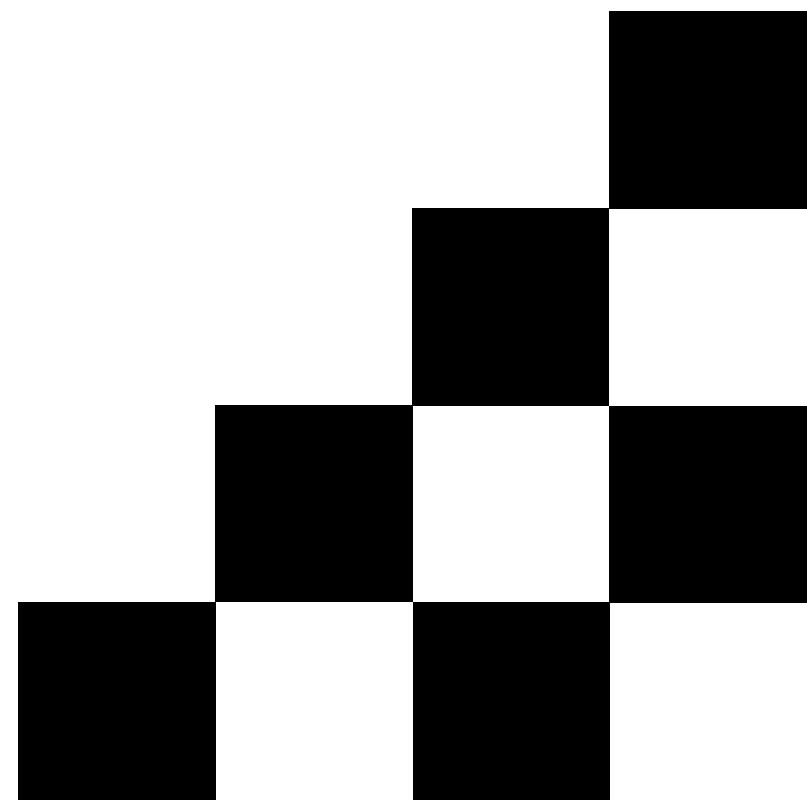
Масштабування

Масштабування - процес збільшення потужностей, на яких запущено систему.

Вертикальне масштабування - збільшення потужності сервера (CPU, RAM, диск).

Горизонтальне масштабування - збільшення кількості серверів і збільшення кількості екземплярів (копій) системи.

Типи нереляційних СУБД (NoSQL)



Документні СУБД

**Документні СУБД зберігають дані у форматі документів -
зазвичай JSON об'єкти.**

**Кожен документ незалежний і може мати різну структуру,
різну вкладеність і тд.**

Можна створювати індекси на поля JSON документів.

Документні СУБД

Переваги:

- Відсутність схеми
- Підходять для глибоко ієрархічних даних
- Дані максимально денормалізовані - швидка вибірка за ключем

Документні СУБД

Недоліки:

- Відсутність схеми
- Дублювання даних через денормалізацію
- Слабша консистентність даних

Документні СУБД

Приклади СУБД:

- MongoDB
- CouchDB
- Firestore
- DocumentDB

Ключ-Значення СУБД

Ключ-Значення СУБД зберігають пари ключ-значення, за своєю суттю відповідають асоціативним масивам.

Ключ - рядок, значенням може мати різний тип даних.

Пошук виключно за ключем.

Ключ-Значення СУБД

Переваги:

- Надзвичайно швидкий запис і читання**
- Легке масштабування**
- Дуже проста модель**

Ключ-Значення СУБД

Недоліки:

- Читання лише за повним ключем**
- Підтримує лише найпростіші операції - get, put, delete**

Ключ-Значення СУБД

Приклади СУБД:

- Redis
- Hazelcast
- Riak

Column-Family СУБД

Column-Family (сімейство колонок) СУБД зберігають набори колонок (таблиці).

Відмінність від реляційних СУБД в тому, що відсутні будь-які зв'язки між таблицями та немає підтримки SQL.

Мають чітку схему.

Column-Family СУБД

Переваги:

- Надзвичайно швидкий запис і читання
- Майже лінійне міштабування
- Конфігурований рівень консистентності даних
- Можуть працювати з неймовірними об'ємами даних

Column-Family СУБД

Недоліки:

- Читання даних лише за повним ключем
- Дуже низька гнучкість запитів
- Схема формується навколо запитів
- Складно розвивати систему, що використовує даний вид СУБД

Column-Family СУБД

Приклади СУБД:

- Cassandra
- ScyllaDB
- HBase

Графові СУБД

Графові СУБД зберігають дані у вигляді графу - вершини + зв'язки.

**Зв'язки між вершинами можуть мати типи та властивості.
Використовують особливі мови (не SQL) для роботи з
графовими даними.**

Графові СУБД

Переваги:

- Ідеально підходять для даних з зв'язками
- Висока швидкодія роботи зі зв'язками
- Надзвичайно гнучна робота зі зв'язками

Графові СУБД

Недоліки:

- Спеціалізований тип СУБД, що не підходить для більшості випадків
- Складно масштабуються

Графові СУБД

Приклади СУБД:

- Neo4j
- Amazon Neptune

Пошукові рушії

СУБД, що спеціалізовані для повнотекстового пошуку.
Мають набір інструментів для ефективного пошуку як за окремими колонками, так і для ефективного повнотекстового пошуку по великих масивах даних та гнучкого ранжування результатів.
Працює на основі розбирання тексту на окремі токени і формування індексів документів на основі цих токенів.

Пошукові рушії

Переваги:

- Ефективний повнотекстовий пошук
- Висока гнучкість ранжування результатів

Пошукові рушії

Недоліки:

- Індексація тексту може займати час, текст не одразу після вставки доступний для пошуку
- Можуть бути примхливими та русурсоємними, підтримка часто проблемна

Пошукові рушії

Приклади:

- ElasticSearch
- Apache Solr
- OpenSearch

Колонкові СУБД

Зберігають дані у форматі таблиць, проте кожна колонка зберігається в окремому файлі.

Не мають зв'язків та обмежень, проте підтримують більшу частину стандарту SQL.

Ефективно працюють з неймовірними об'ємами даних.

Колонкові СУБД

Переваги:

- Ефективні для аналітичних SQL запитів
- Дуже гнучко масштабуються
- Працюють з дуже великими об'ємами даних

Колонкові субд

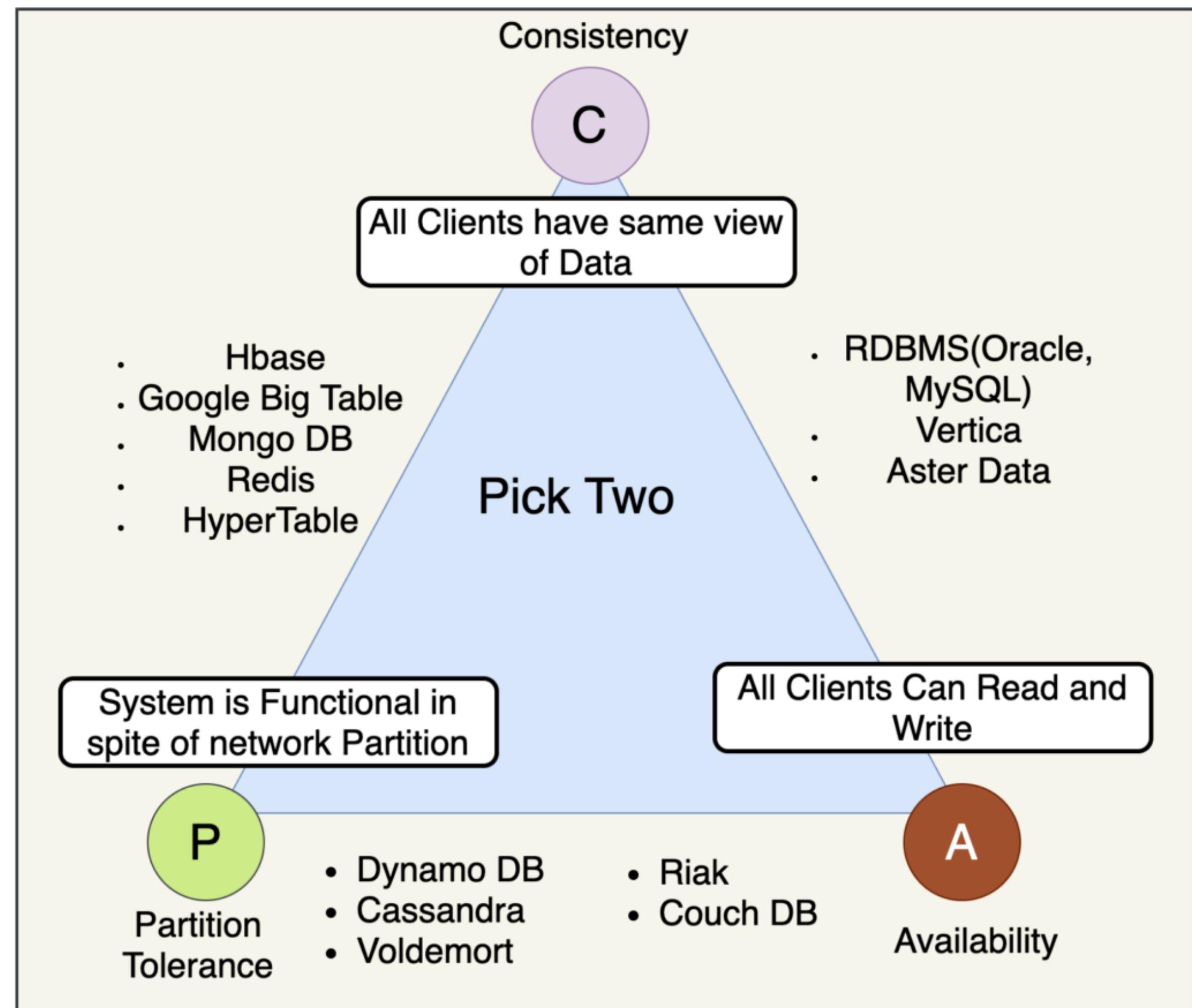
Переваги:

- Висока вартість
- Відсутність зв'язків та обмежень
- Низька швидкодія OLTP запитів

Колонкові СУБД

Приклади СУБД:

- Snowflake
- Amazon Redshift
- Clickhouse



Варто обирати реляційні СУБД коли

- Необхідна консистентність та ACID
- Складні запити
- Необхідні зв'язки
- Консистентність важливіша за доступність
- Немає необхідності використовувати нереляційну СУБД

Варто обирати нереляційну СУБД коли

- **Необхідне горизонтальне масштабування**
- **Висока кількість записів**
- **Вимоги до надвисокої швидкодії**
- **Доступність важливіша за консистентність**
- **Необхідність використання спеціалізованого
типу СУБД**

Рекомендації щодо вибору

- Завжди починати з реляційної СУБД
- Мінімізувати кількість різних СУБД для проекту
- Вийняток - ключ-значення СУБД
- Переходити на нереляційні СУБД, лише якщо є необхідність

Запитання