

NoSQL базы данных

NoSQL, Hbase, Cassandra

- Реляционные базы данных
- NoSQL
- Hbase
- Cassandra

Способы хранения данных

- Память
- Файлы
- Базы данных

Память: плюсы

- + Широкий выбор структур данных
- + Возможность создавать свои типы данных
- + Быстрый доступ к данным: чтение, изменение, дополнение
- + Высокая скорость

Память: минусы

- Размер данных ограничен оперативной памятью
 - Данные существуют, пока жив процесс
 - Нужно реализовать механизмы одновременного доступа
-
- Низкая надежность

Память: примеры

- Игры на Dendy
- Компиляторы
- Word, Excel
- IDE (интегрированная среда разработки)

Файлы: плюсы

- + Существенно больший объем данных
- + Свобода в формате и структуре данных
- + Простые механизмы доступа к данным в файле
- + Отсутствие третьей стороны при работе с данными

Файлы: минусы

- Сложно вносить изменения в файл
- Медленный доступ к данным
- Тяжело организовать совместный доступ к данным
- Отсутствие контроля целостности данных
- Сложные механизмы доступа к данным, расположенным в нескольких файлах

Файлы: примеры

- Игры на ПК (нет одновременного доступа)
- Word, Excel (не нужен произвольный доступ)
- Поисковый индекс (только операция чтения)

Реляционная Модель Данных (РМД)

- Структурный аспект
- Аспект целостности
- Аспект обработки

Объекты РМД

A_1	...	A_N

Реляционные Базы Данных: ACID

Atomicity – Атомарность

Consistency – Согласованность

Isolation – Изолированность

Durability – Надежность

РБД: плюсы

- + Универсальный доступ к данным (язык SQL)
- + Контроль за целостностью данных (ACID)
- + Одновременный доступ к данным
- + Повышенная безопасность

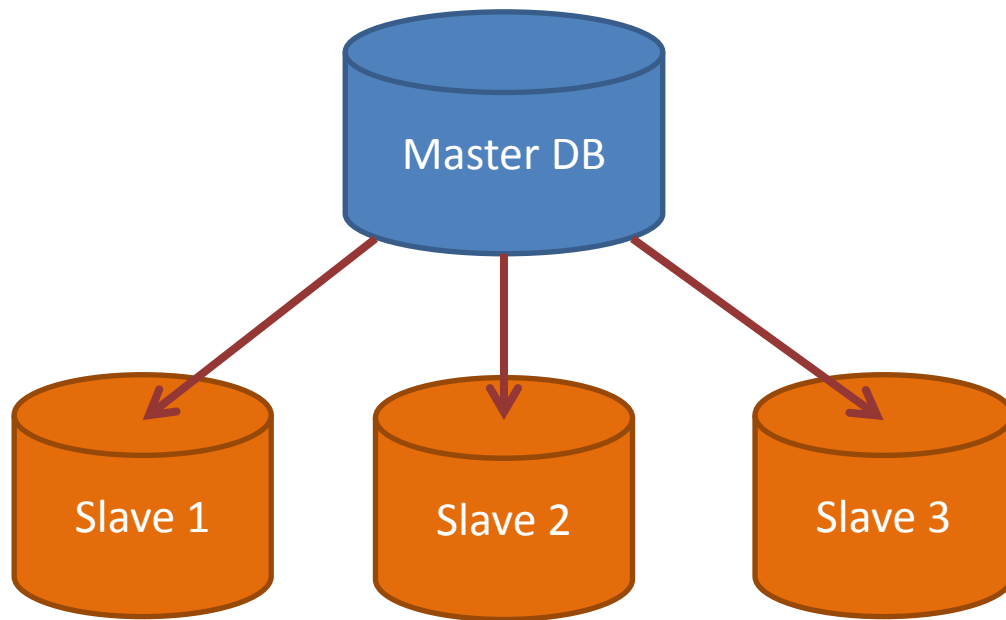
РБД: минусы

- Тяжело хранить иерархические данные
- Проблемы с масштабируемостью

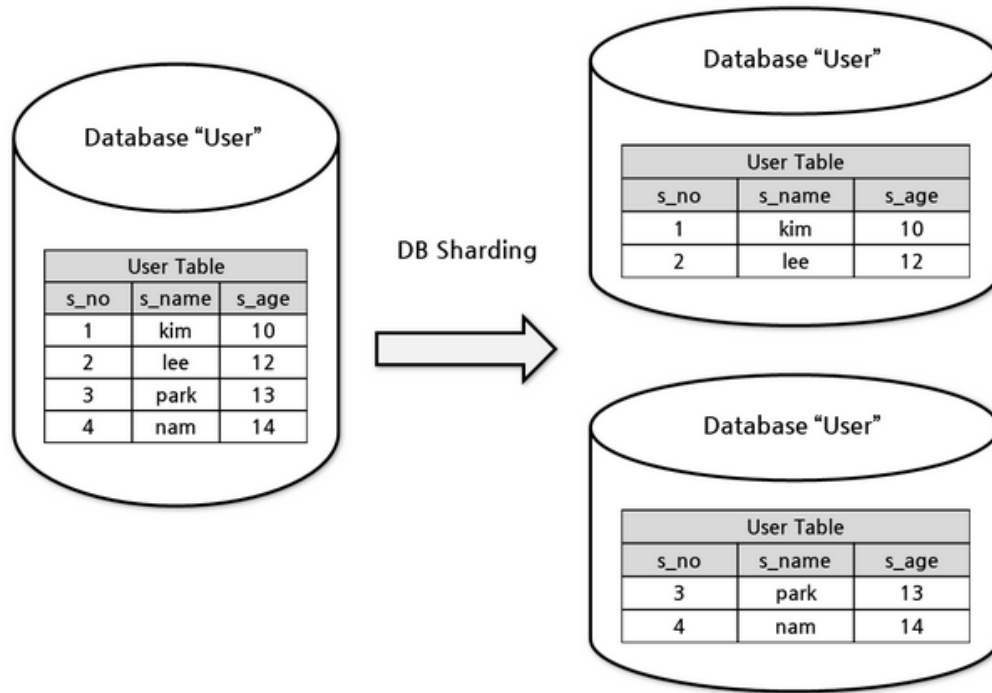
РБД: примеры

- Системы документооборота
- Интернет магазины
- Социальные сети
- Базы данных гос. органов (налоговая, миграционная, МВД, ...)

Масштабирование РБД: Master/Slave



Масштабируемость РБД: Sharding



Not only SQL

Основные черты NoSQL

- Применение различных типов хранилищ
- Нефиксированная схема БД
- Использование многопроцессорности
- Линейная масштабируемость
- Сокращение времени разработки

Базовые события в становлении NoSQL

- BigTable (Google)
- Dynamo (Amazon)
- CAP Theorem

Основные свойства системы

- ***Consistency*** – непротиворечивость данных
- ***Availability*** – доступность данных
- ***Partitionability*** – разделяемость данных на изолированные части

CAP Theorem

Теорема: Можно иметь только два из трех свойств в любой *shared-data* системе

Consistency

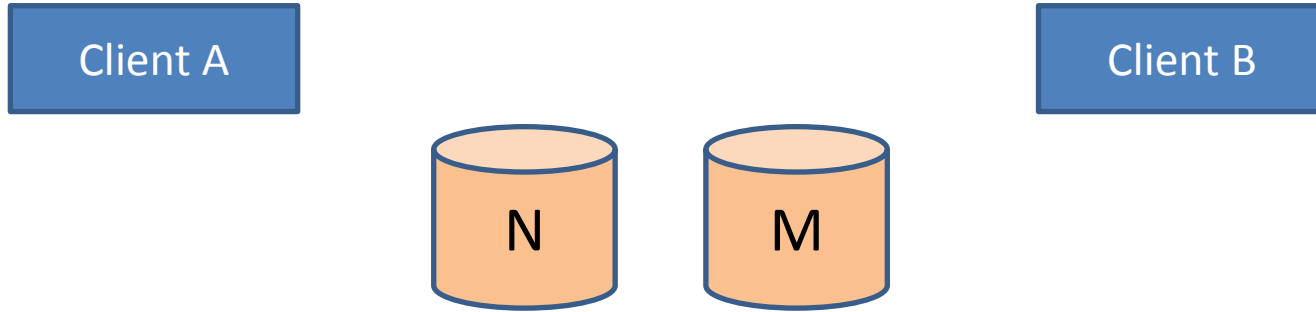
Availability

Partitionability

Consistency Models

- Строгая (*strict consistency*)
- Последовательная (*sequential consistency*)
- Причинная (*causal consistency*)
- Процессорная (*processor consistency*)
- Слабая (*weak consistency*)
- **Консистентность в конечном счете (*eventual*)**
- Консистентность по выходу (*release consistency*)
- Консистентность по входу (*entry consistency*)

Consistency Model: пример



Видит ли клиент В записанные данные от А?

- Strict Consistency: *Да*
- Eventual consistency: *Может быть*

Consistency Model: пример

CAP теорема говорит: *Strict Consistency* не может быть достигнута одновременно с *availability* и *partition-tolerance*

BASE вместо ACID

BASE:

- *Basically Available* – базовая доступность
- *Soft State* – гибкое состояние
- *Eventually Consistent* – согласованность в конечном счете

Типы NoSQL

- **Key/Value** (модель данных: *хеш-таблица*)
 - Amazon S3 (Dynamo)
 - Voldemort
- **Column-based** (модель данных: *разряженная матрица*)
 - HBase
 - Cassandra

Типы NoSQL

- **Document-based** (модель данных: *дерево*)
 - MongoDB
 - OrientDB
- **Graph-based** (модель данных: *граф*)
 - Allegro
 - InfiniteGraph

Key/Value

- + Высокая скорость
- + Хорошая масштабируемость
- + Простая модель данных
- Многие структуры данных сложно представить в виде key/value

Column-based

- + Более богатые структуры данных
- Меньшая согласованность данных
- Хуже масштабируется