

Типы и структура СУБД



Тора
Самисько



Тора Самисько

GameDev, OAI

План занятия

1. [Базы Данных](#)
2. [Системы управления БД](#)
3. [CAP-теорема](#)
4. [PACELC-теорема](#)
5. [Транзакции](#)
6. [ACID](#)
7. [BASE](#)
8. [NoSQL](#)
9. [MongoDB](#)
10. [Redis](#)
11. [Memcached](#)
12. [Итоги](#)
13. [Домашнее задание](#)



Базы Данных

Базы Данных

Имеется множество определений термина “База данных” (БД). Нет “идеального” описания термина.

Ключевые моменты в каждом описании термина “БД”:

- БД хранится и обрабатывается вычислительной системой
- Данные в БД логически структурированы
- БД включает схему или метаданные, описывающие её структуру

Первый момент является строгим, остальные допускают различные трактовки.



Системы управления БД

Системы управления БД

Система управления базой данных (СУБД) - это ПО, предназначенное для:

- определения,
- обработки,
- извлечения,
- управления

данными в базе данных.

СУБД обычно управляет самими данными, форматом данных, именами полей, структурой записи и структурой файлов.

Также определяет правила для проверки и управления этими данными.

Системы управления БД

СУБД можно разделить на следующие категории:

- **Реляционные**
- Объектно-ориентированные
- **NoSQL**
 - Иерархические
 - Графовые
 - Сетевые
 - **Документо-ориентированные**
 - **Ключ-значение**
 - Column-oriented

Системы управления БД

Реляционные

В системах управления реляционными базами данных (СУБД) отношения между данными являются реляционными, и данные хранятся в виде таблиц.

Каждый столбец таблицы представляет атрибут, а каждая строка в таблице представляет собой запись.

Каждое поле в таблице представляет собой значение данных.

Используют SQL для взаимодействия с данными.

OrderItem			Product	
order_id	product_id	quantity	id	name
100	55	2	55	Iphone 5
100	56	1	56	Ipad 3

Системы управления БД

Объектно-ориентированные

Предоставляют полнофункциональные возможности программирования баз данных, сохраняя при этом совместимость с ООП языком.

Добавляет функциональность базы данных в ООП ЯП.



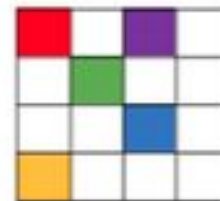
Системы управления БД

NoSQL

Базы данных NoSQL не используют SQL в качестве основного языка доступа к данным.

NoSQL не имеет predetermined схем, что делает её идеальным кандидатом для быстро меняющихся сред разработки.

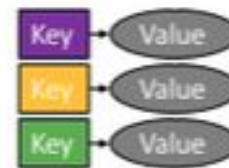
NoSQL позволяет разработчикам вносить изменения «на лету», не затрагивая приложения.



Column



Graph



Key-Value



Document

Системы управления БД

Иерархические

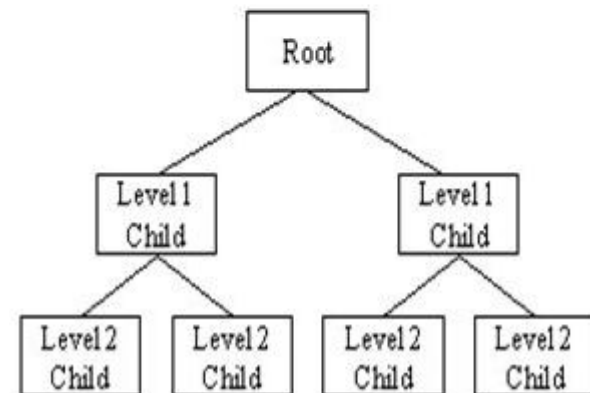
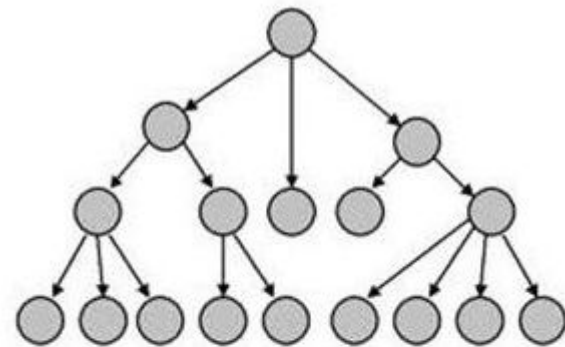
В иерархической модели данные организованы в древовидную структуру.

Данные хранятся в виде набора полей, где каждое поле содержит только одно значение.

Записи связаны друг с другом через связи в отношениях родитель-потомок.

В иерархической модели базы данных каждая дочерняя запись имеет только одного родителя.

Родитель может иметь несколько детей.



Системы управления БД

Графовые

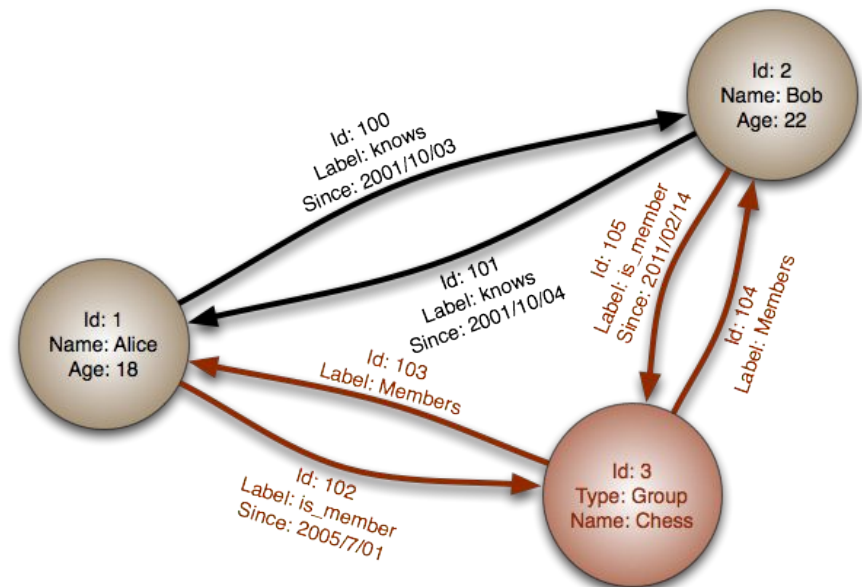
NoSQL БД, которая используют структуру графов для семантических запросов.

Данные хранятся в виде узлов, ребер и свойств.

Узел представляет собой объект.

Ребро представляет собой отношение, которое соединяет узлы.

Свойства - это дополнительная информация, добавляемая к узлам.



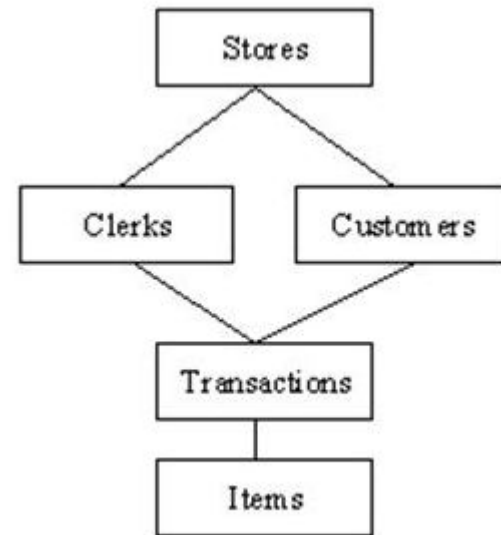
Системы управления БД

Сетевые

Сетевые СУБД используют сетевую структуру для создания отношений между объектами.

Сетевые базы данных - имеют иерархическую структуру, но в отличие от иерархических баз данных, где у одного узла может быть только один родительский узел, сетевой узел может иметь отношения с несколькими объектами.

Сетевая база данных больше похожа на “паутину”.

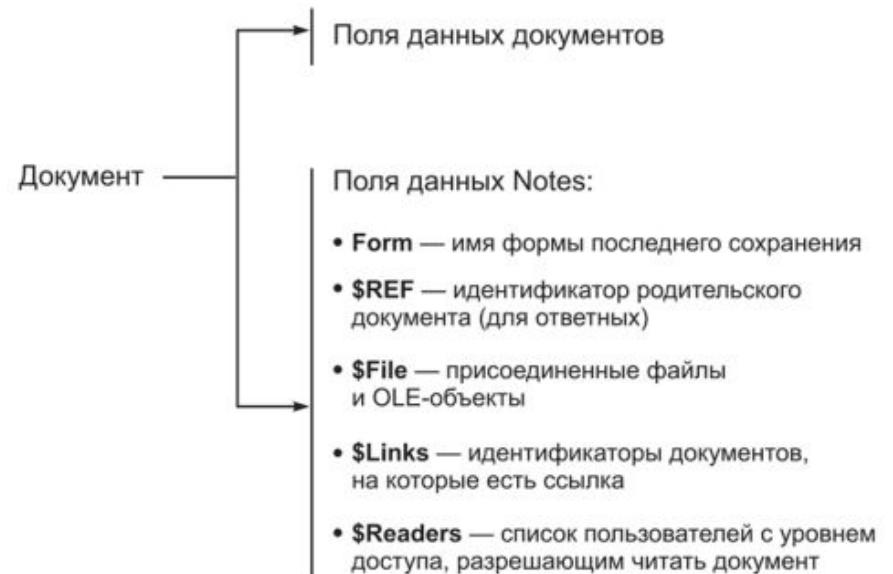


Системы управления БД

Документо-ориентированные

Является NoSQL БД, в которой данные хранятся в виде документов.

Каждый документ представляет данные в виде ключ-значение, связь с другими документами и мета-полями.



Системы управления БД

Ключ-значение

База данных на основе пар «ключ-значение» хранит данные как совокупность пар «ключ-значение», в которых ключ служит уникальным идентификатором.

Как ключи, так и значения могут представлять собой что угодно: от простых до сложных составных объектов.

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Системы управления БД

Column-oriented

В таких системах данные хранятся в виде матрицы, строки и столбцы которой используются как ключи.

Типичным применением этого типа СУБД является веб-индексирование, а также задачи, связанные с большими данными, с пониженными требованиями к согласованности.

Каждая строка имеет свой набор столбцов.

row key	columns ...			
jbellis	name	email	address	state
	jonathan	jb@ds.com	123 main	TX
dhutch	name	email	address	state
	daria	dh@ds.com	45 2 nd St.	CA
egilmore	name	email		
	eric	eg@ds.com		



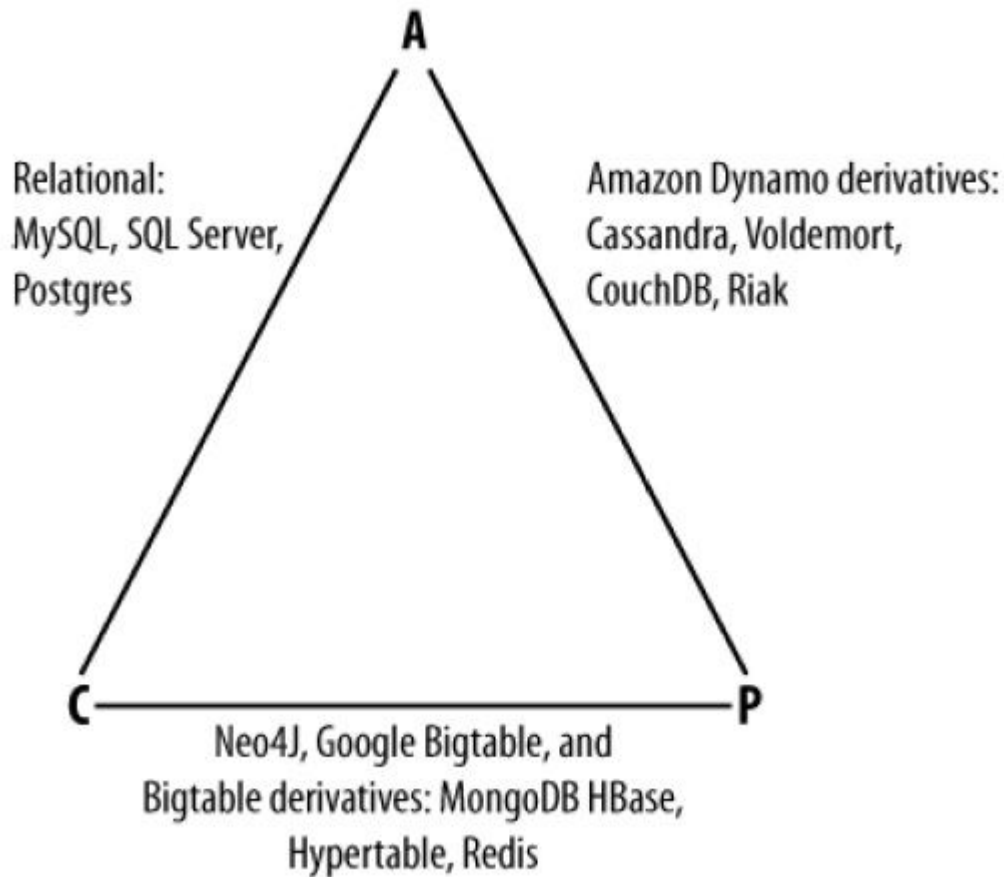
CAR-теорема

CAP-теорема

CAP-теорема (теорема Брюера) - утверждение о том, что в любой реализации распределённых вычислений возможно обеспечить не более двух из трёх следующих свойств:

- **согласованность данных (англ. consistency)** — во всех вычислительных узлах в один момент времени данные не противоречат друг другу;
- **доступность (англ. availability)** — любой запрос к распределённой системе завершается корректным откликом, однако без гарантии, что ответы всех узлов системы совпадают;
- **устойчивость к разделению (англ. partition tolerance)** — расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

CAP-теорема



CAP-теорема

Недостатки:

- **Условность понятий CAP**
Например, система может отвечать в течении часа - если ответ корректный, в рамках CAP теоремы, это доступная система.
- **В основном, все системы - CP и AP**
Сетевые взаимодействия допускают обрывы связи и потери пакетов - вследствие этих накладных расходов нельзя гарантировать CA.
- **Множество систем удовлетворяют только P**
В Master-Slave системе при потере Master - теряется CA. В асинхронной Master-Slave системе запрос данных может производиться раньше синхронизации всех Slave.
- **Сложность применения к NoSQL**



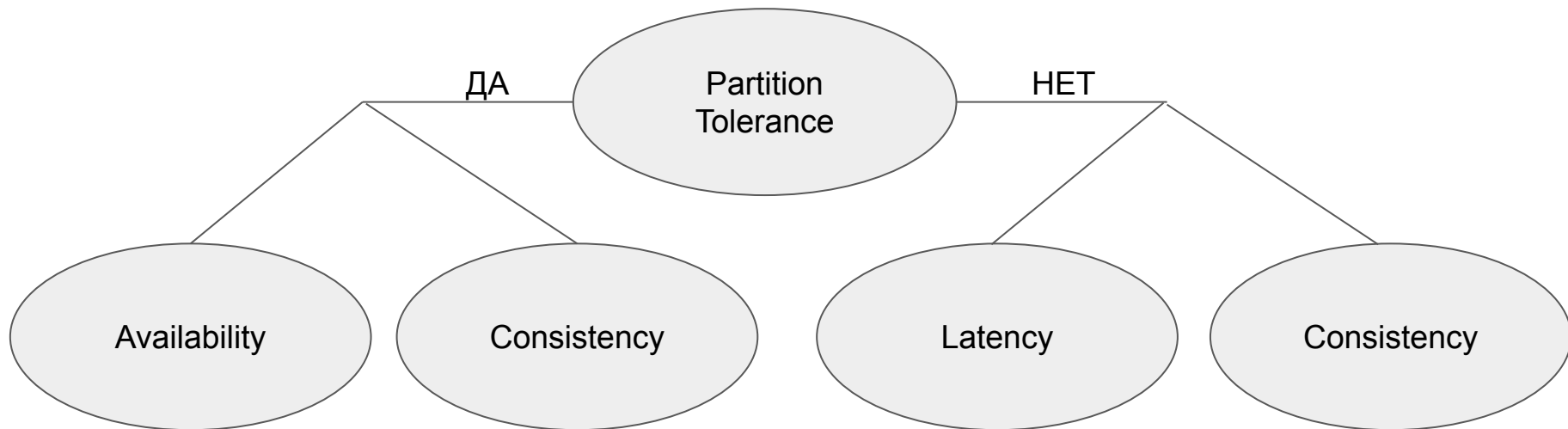
PACELC-теорема

PACELC-теорема

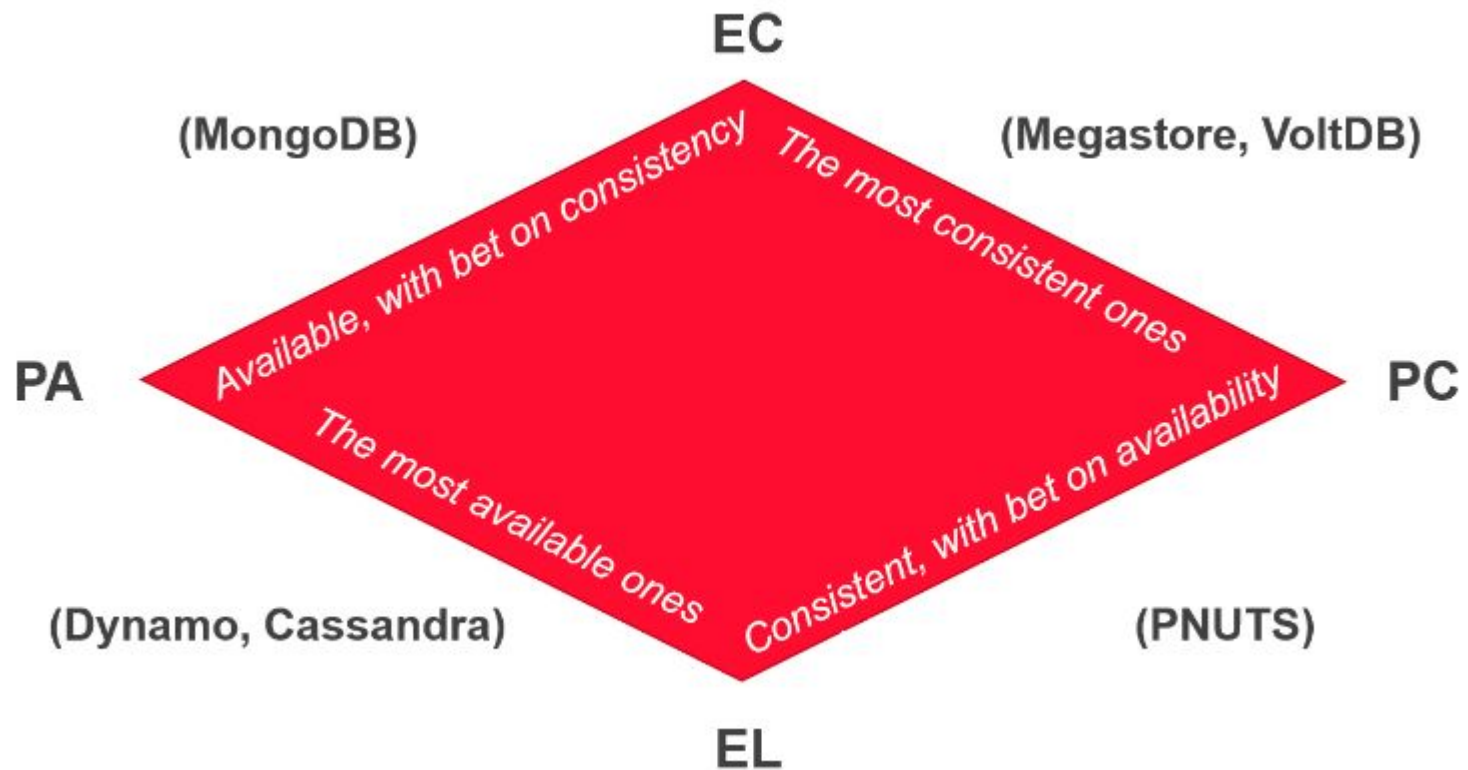
Расширение CAP-теоремы.

Добавляет понятие Latency - время, за которое клиент получит ответ и которое регулируется каким-либо уровнем согласованности.

При расчете, сводится к виду:



PACELC-теорема





Транзакции

Транзакции

Транзакция - это набор последовательных операций над БД, представляющих логическую единицу.

Транзакция применяется полностью или не применяется совсем.

Пример:

Необходимо перевести с банковского счёта номер 1 на счёт номер 2 сумму в 10 рублей.

Этого можно достичь, приведённой последовательностью действий (транзакцией):

1. Прочитать баланс на счёту номер 1.
2. Уменьшить баланс на 10 рублей.
3. Сохранить новый баланс счёта номер 1.
4. Прочитать баланс на счёту номер 2.
5. Увеличить баланс на 10 рублей.
6. Сохранить новый баланс счёта номер 2.



ACID

ACID

ACID - требования к СУБД, в обеспечение надежности и предсказуемости ее работы.

- **A - atomicity (атомарность)** никакая транзакция не будет зафиксирована в БД частично.
- **C - consistency (согласованность)** каждая успешная транзакция фиксирует только допустимые результаты.
- **I - isolation (изоляция)** параллельные транзакции не искажают результат друг друга.
- **D - durability (стойкость)** гарантия применения успешных транзакций, независимо от низкоуровневых проблем.

ACID позволяет проектировать высоконадежные системы.



BASE

BASE

BASE - принцип, противопоставляющий себя ACID.

- **BA - basically availability (базовая доступность)** деградация части узлов ведет к деградации части сессий, исключая полную деградацию системы. Система отвечает на любой запрос, но в ответе могут быть неверные данные.
- **S - soft state (неустойчивое состояние)** уменьшение времени хранения сессий и фиксация обновлений только критичных операций.
- **E - eventually consistent (конечная согласованность)** изменение состояния в конечном итоге применится на все системы.

BASE позволяет проектировать высокопроизводительные системы.



NoSQL

NoSQL

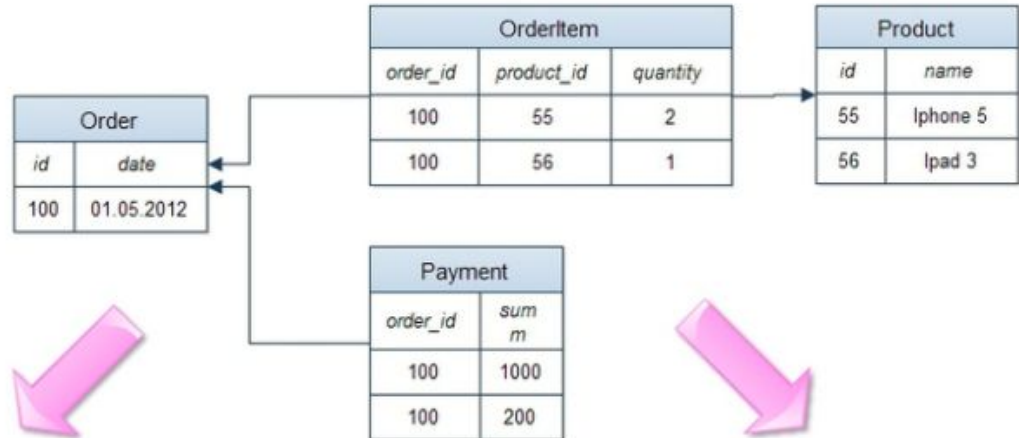
NoSQL - огромное семейство БД, полный список всех систем можно прочитать на сайте: hostingdata.co.uk/nosql-database/

Общие характеристики NoSQL систем:

- No SQL - Не используется SQL (в классическом виде).
- Schemaless - Данные не структурированы.
- Aggregates - Данные представлены в виде агрегатов.
- BASE - Слабые ACID свойства, уклон в сторону BASE для производительности.
- Share nothing - NoSQL распределенные системы, без совместно используемых ресурсов.

NoSQL

Relational model



Aggregate model 1

```
// Order document
{
  "id": 100,
  "customer_id": 1000,
  "date": 01.05.2012,
  "order_items": [
    {
      "product_id": 55,
      "product_name": Iphone5,
      "quantity": 2
    },
    {
      "product_id": 56,
      "product_name": Ipad3,
      "quantity": 1
    }
  ],
  "payments": [
    {
      "sum": 1000,
      "date": 03.05.2012
    }
  ]
}
// Product document here
{...}
```

Aggregate model 2

```
// Order document
{
  "id": 100,
  "customer_id": 1000,
  "date": 01.05.2012,
  "order_items": [
    {
      "product_id": 55,
      "product_name": Iphone5,
      "quantity": 2
    },
    {
      "product_id": 56,
      "product_name": Ipad3,
      "quantity": 1
    }
  ]
}
// Payment document
{
  "order_id": 100,
  "sum": 1000,
  "date": 03.05.2012
}
// Product document here
{...}
```



MongoDB

MongoDB

MongoDB - одна из популярных документо-ориентированных СУБД. Является классическим примером NoSQL.

MongoDB поддерживает:

- ad-hoc запросы
- Индексирование
- Горизонтальное масштабирование и шардинг
- MapReduce
- Транзакции, ACID/BASE

По PACELC теореме MongoDB соответствует PA/EC.

MongoDB

MongoDB подходит для следующих применений:

- хранение и регистрация событий;
- системы управления документами и контентом;
- электронная коммерция;
- игры;
- данные мониторинга, датчиков;
- мобильные приложения;
- хранилище операционных данных веб-страниц
(например, хранение комментариев, рейтингов, профилей пользователей, сеансы пользователей).



Redis

Redis

Redis - это СУБД вида “ключ-значение”.

Основные характеристики системы:

- Может использоваться как БД, так и как кэш-система или брокер сообщений.
- Все данные хранятся в оперативной памяти.
- Данным можно присваивать Time-To-Live.
- Имеется встроенная система Pub/Sub.
- Поддерживает Master-Slave репликацию.

Важно! “Из коробки” не имеет механизма консенсуса. При отказе ведущей реплики - необходимо вручную выбрать новую ведущую реплику.

Redis

Redis Sentinel - система управления узлами Redis, осуществляющая:

- мониторинг работоспособности ведущих и ведомых узлов;
- алертинг о произошедших отклонениях в работе;
- автоматический выбор нового ведущего узла, в случае отказа текущего;
- механизм нотификации клиентов и узлов о перевыборе ведущего узла.

Redis Sentinel входит в состав Redis начиная с версии 2.6.

Sentinel рекомендуется использовать в режиме кластера для обеспечения его отказоустойчивости.



Memcached

Memcached

Memcached - это СУБД вида “ключ-значение”.

Основные характеристики системы:

- используется как распределенный кэш;
- все данные хранятся в оперативной памяти;
- данным можно присваивать Time-To-Live;
- поддерживает Master-Slave репликацию.

Обладает меньшей функциональностью, по сравнению с Redis



Итоги

Итоги

В данной лекции мы:

- рассмотрели какие бывают СУБД;
- научились определять тип СУБД по CAP/PACELC теоремам;
- ознакомились с понятием транзакций;
- узнали что такое ACID/BASE принципы;
- рассмотрели популярные NoSQL решения и их характеристики.



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** telegram.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Тора Самисько