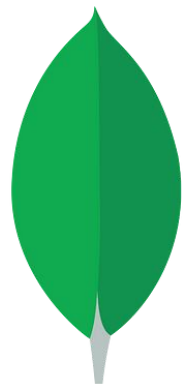


Л/Р №7 по УД  
Доклад о NoSQL решении

Команда: студенты 23КНТ-2  
Виноградов Константин, Дружков  
Матвей, Сапрыкин Николай,  
Сапрыкина Ксения, Субботина  
Софья



mongoDB®

*mongo* — от *humongous*  
(огромный)



Написана на C++

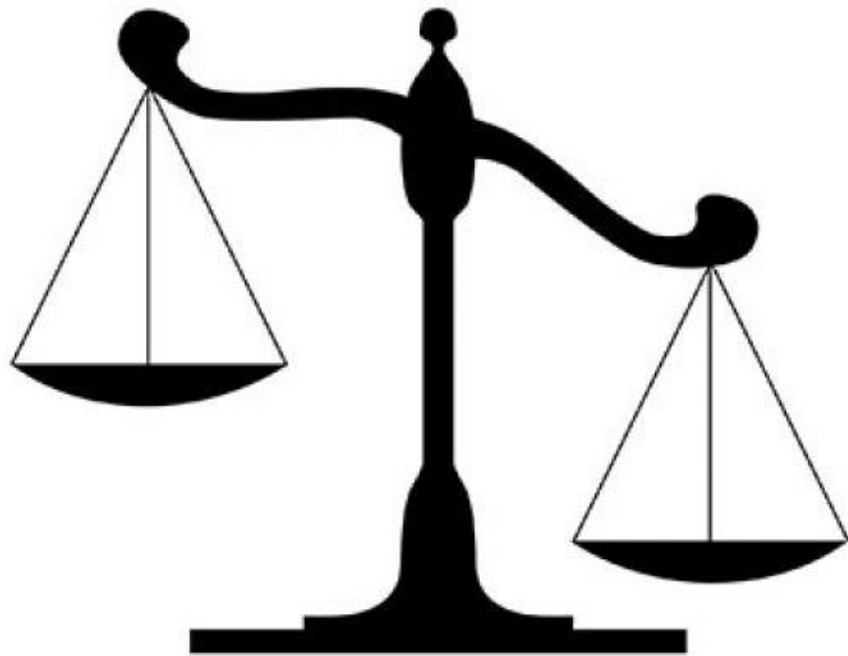
Первый выпуск: 2009

Разработчик: MongoDB (в прошлом — 10gen)



**mongo DB**

*mongoDB* – документо-ориентированная *NoSQL*  
Данные хранятся в виде документов (*BSON*),  
а не строк



данные весят больше, НО скорость их обработки выше

# Архитектура и принципы работы

## Основные концепции MongoDB:

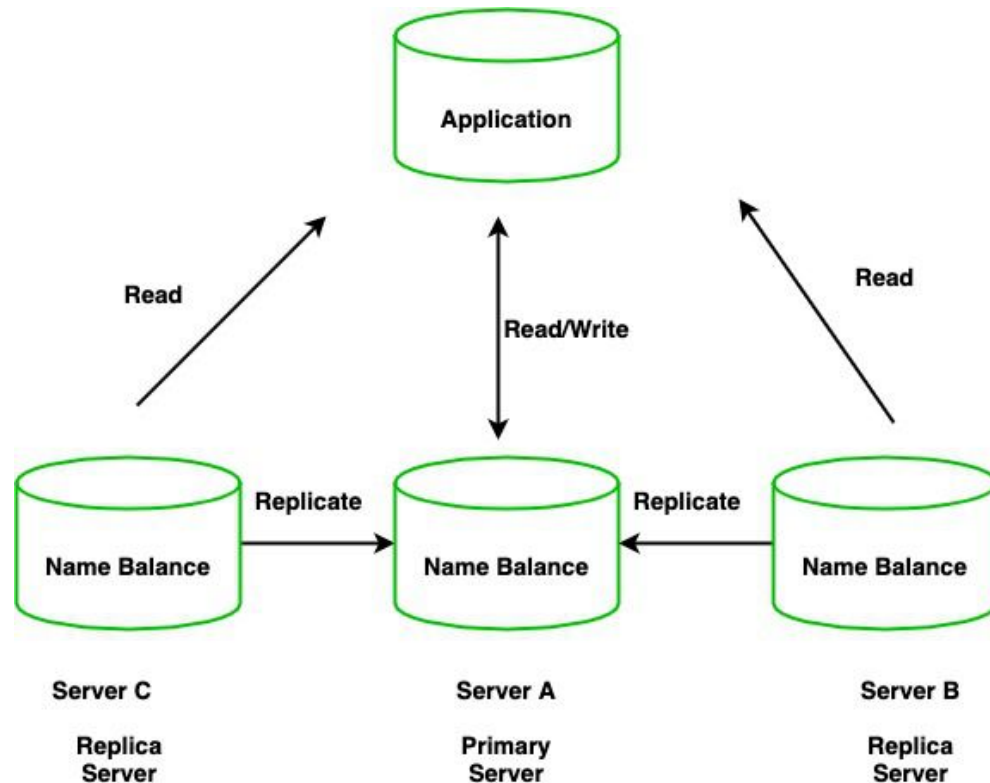
- *Документ* – набор пар “ключ-значение”
- *Коллекция* – аналог таблицы без жесткой схемы
- *\_id* – уникальный идентификатор
- *Гибкость структуры* – можно легко изменять структуру без миграций
- *Отсутствие NULL* – ключ без значения не включается в документ

# Репликация

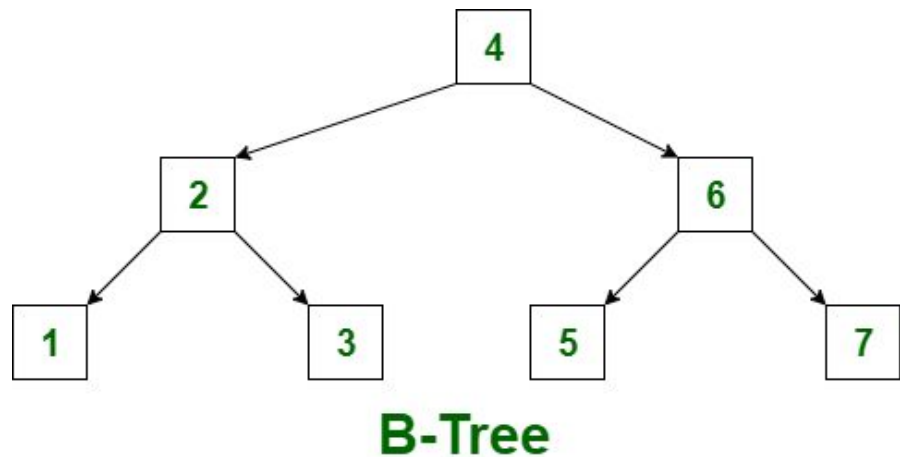
Репликационный набор – один главный узел (*primary*) и несколько вторичных (*secondary*).

В случае отказа главного узла один из вторичных становится главным.

**read preference** – для чтения со вторичных узлов.



# Индексация



Индексы создаются на любом поле документа и по умолчанию хранятся в виде *B-деревьев*.

Повышает производительность запросов.

# Шардинг

Данные распределяются по нескольким серверам (*shard*).

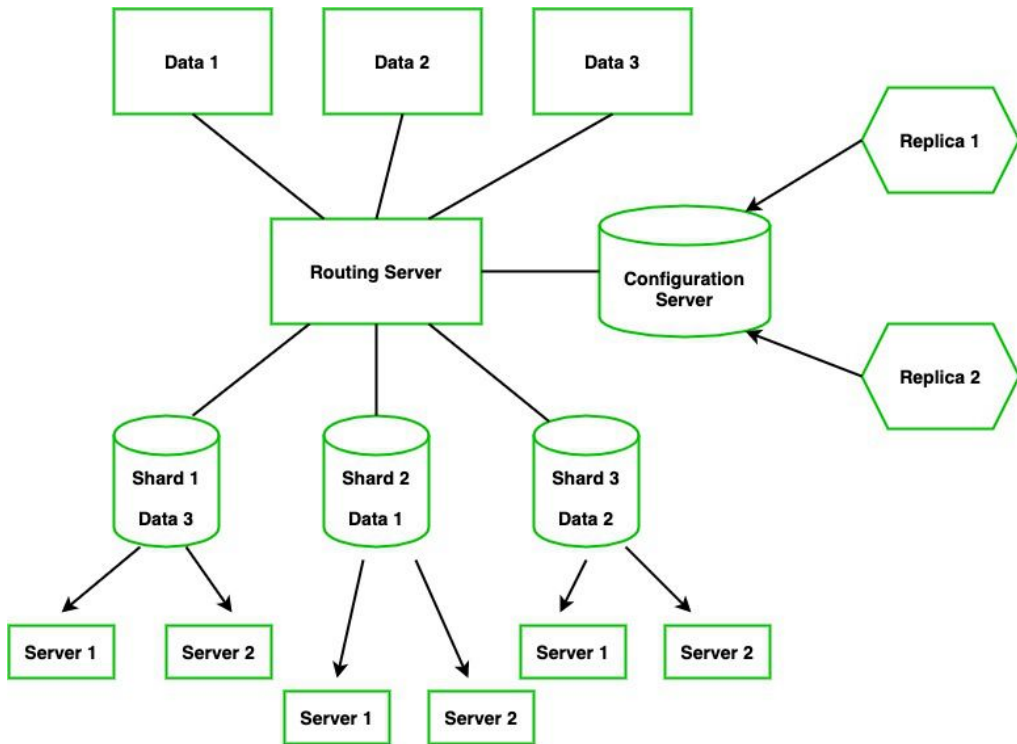
**Balancer** автоматически перераспределяет данные.

**Репликационные наборы** обеспечивают отказоустойчивость.

**Клиенты** взаимодействуют с базой через *mongos*.

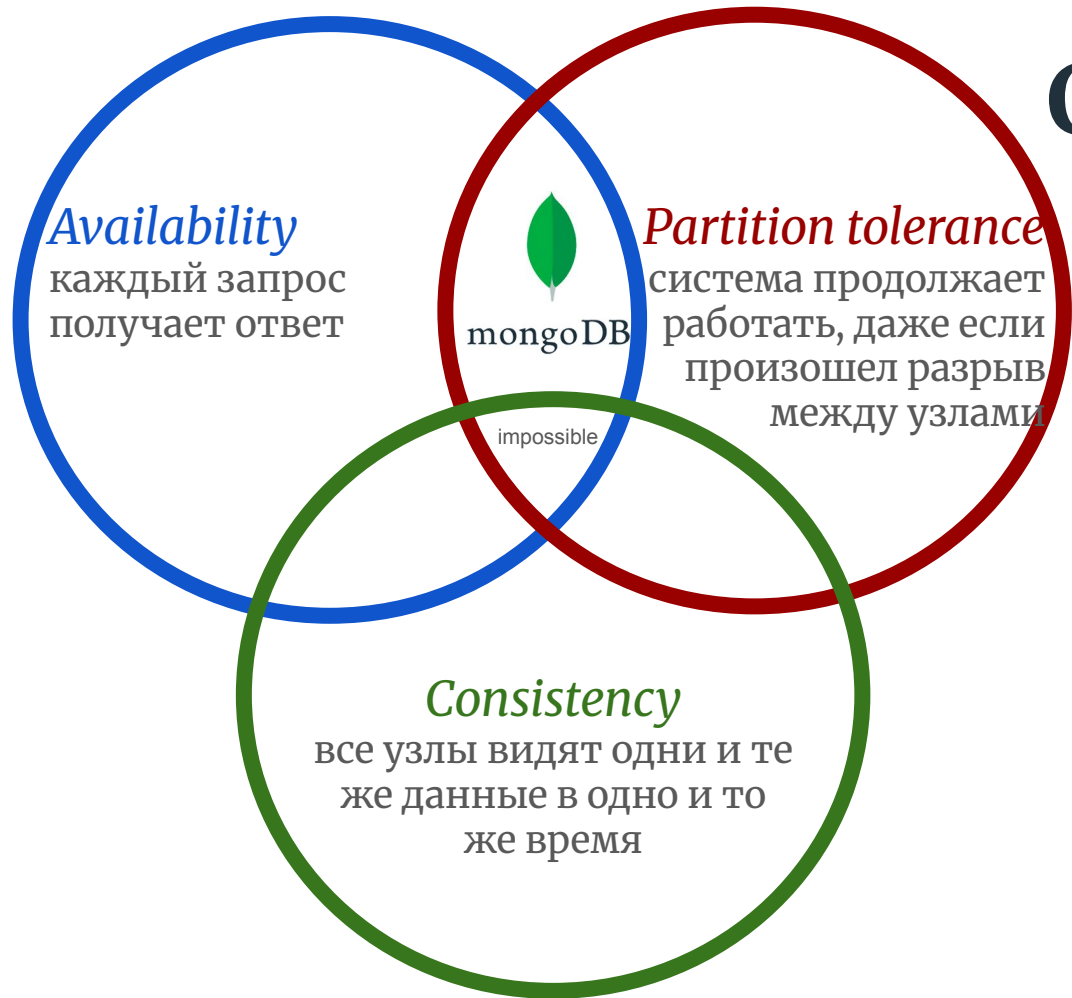
+ высокая масштабируемость

- сложность управления





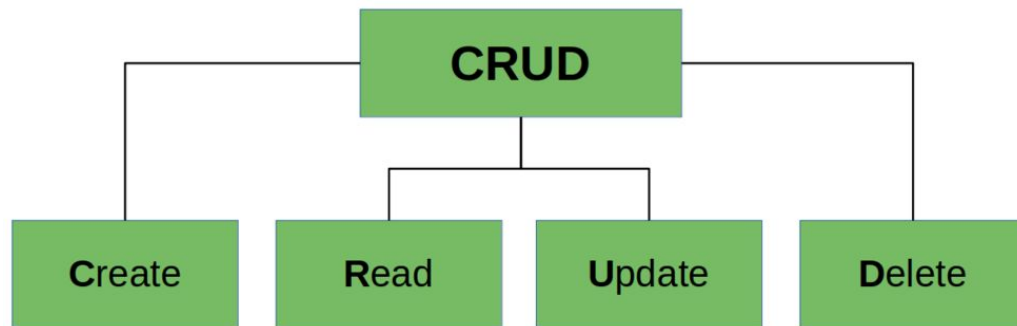
# CAP - теорема



# Функциональные ВОЗМОЖНОСТИ

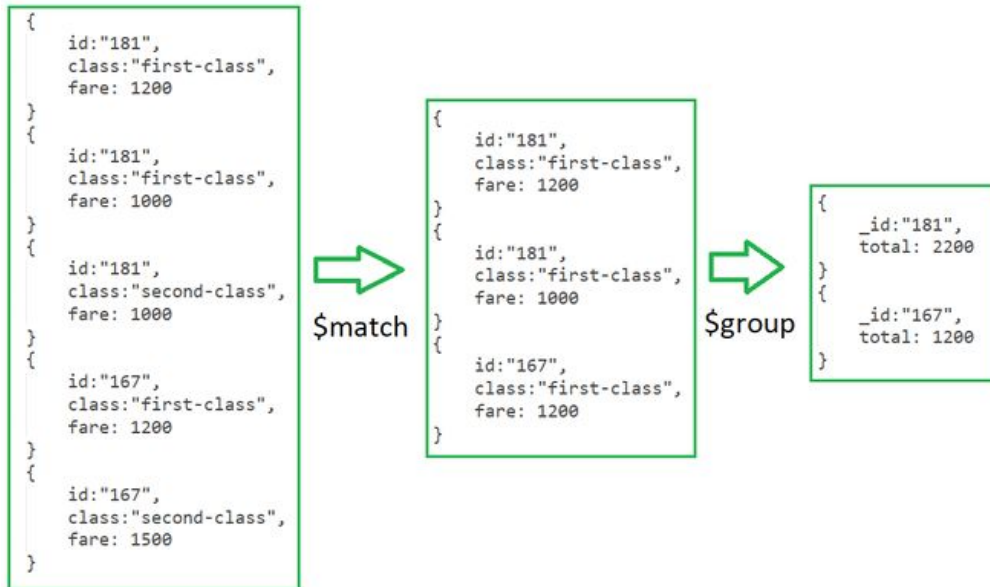
# CRUD-операции

Запросы используют *JSON*-подобный синтаксис.



# Агрегация данных

```
db.train.aggregate( [
  { $match: { class: "first-class" } },
  { $group: { _id: "id", total: { $sum: "$fare" } } } ] pipeline stages
)
```



*Aggregation Pipeline*  
(аналог *GROUP BY*)

*\$match* – фильтрация

*\$group* – группировка

*\$sort* – сортировка

*\$project* – проекция

# Поиск и индексация

## Полнотекстовый поиск

*(Text Search)*

позволяет выполнять запросы  
по текстовым данным.

Геопространственные запросы  
используются для работы с  
картографическими данными.

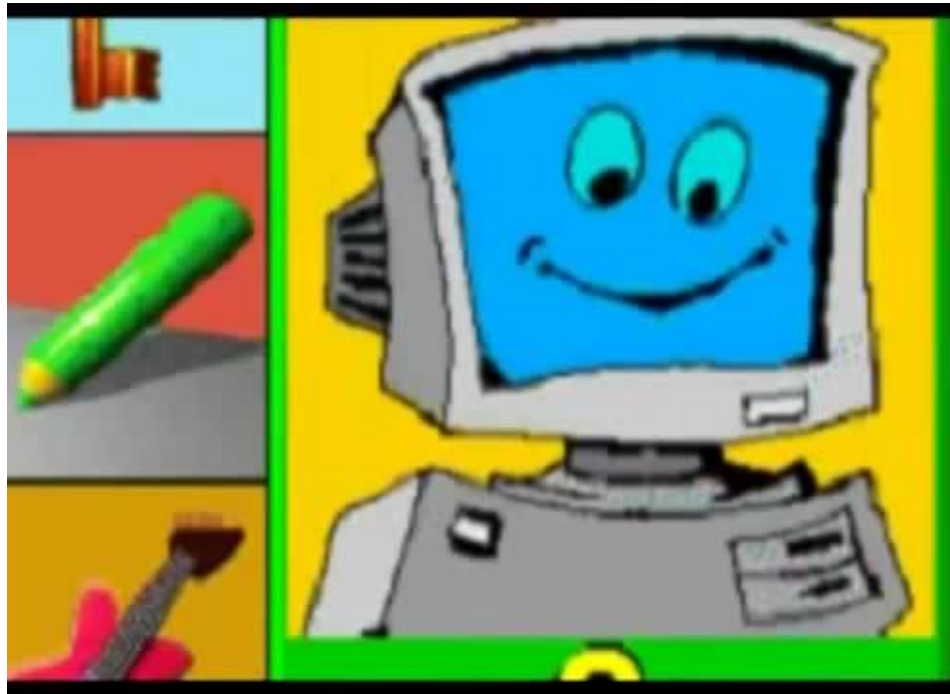
# Хранение больших данных

Ограничение на размер одного документа – 16 МБ.

Для хранения больших файлов используется *GridFS*.

Коллекция *files* – хранит метаданные файлов.

Коллекция *chunks* – хранит фрагменты данных (256 КБ).



# Сравнение с SQL

Функция	SQL	MongoDB
Хранение данных	Структурированные таблицы	Документы BSON
Жесткость схемы	Обязательное определение схемы	Гибкая структура
Индексация	В-деревья, полнотекстовый поиск	В-деревья, полнотекстовый поиск
Связи между таблицами/документами	JOIN (строгие связи)	Вложенные документы, \$lookup (аналог JOIN)
Транзакции	ACID, сложные транзакции	Поддержка multi-document транзакций (но менее мощных)
Масштабирование	Вертикальное (мощное оборудование)	Горизонтальное (шардинг)
Производительность	Высокая для сложных аналитических запросов	Оптимизирована для оперативных данных

## Преимущества mongoDB

+ гибкость  
 + высокая масштабируемость  
 + эффективная работа с неструктурированными данными

## Недостатки mongoDB

– сложности с поддержанием целостности данных  
 – менее эффективные аналитические запросы



# Сравнение с другими NoSQL

По сравнению с  
ключ-целостными  
БД MongoDB сложнее,  
но предлагает больше  
возможностей для  
структурированных  
данных.



В отличие от  
колоночных БД,  
mongoDB проще в  
использовании с  
динамической  
схемой данных.

Документо-ориентированная  
структура делает MongoDB  
более удобной для хранения  
сложных и вложенных  
данных по сравнению с  
графовыми базами данных.



Интеграция с  
другими инструментами

# Внешние системы

- *Драйверы* — доступны для Python (PyMongo), Java, Node.js и других языков.
- *Графические утилиты:*
  - MongoDB Compass — графический интерфейс для работы с базой.
  - mongo shell — CLI для выполнения запросов.
- *BI Connector* — позволяет выполнять запросы через BI-инструменты.
- *Dremio, Presto* — SQL-движки, работающие с MongoDB.
- *MongoDB Atlas* — облачное решение с API.

# Подключение к **mongoDB**



**mongo DB**

- Для *драйверов*: установка библиотеки и указание строки подключения.
- Для *Compass*: ввод параметров подключения.
- Для *Atlas*: регистрация в облаке и настройка *API*-токена.

# Сравнение методов доступа

Метод	SQL	MongoDB
JDBC/ODBC	Да	BI Connector (аналог)
Прямой API-доступ	REST, GraphQL	Драйверы (PyMongo, Node.js, Java)
Визуальные инструменты	pgAdmin, MySQL Workbench	Compass, MongoDB Atlas
Облачные решения	AWS RDS, Azure SQL	MongoDB Atlas

Спасибо за внимание!