

Predictive maintenance

Лекция 4
распределённые хранилища данных

Власов Кирилл Вячеславович



Hive: SQL Поверх больших данных



Hive: SQL Поверх больших данных



Hive – это инструмент инфраструктуры хранилища данных для обработки структурированных данных в Hadoop.



ORACLE®
D A T A B A S E

 PostgreSQL

The PostgreSQL logo features a blue icon of a stylized elephant's head facing left, positioned to the left of the word "PostgreSQL".



*Обычные SQL БД не
достаточно гибкие в
масштабировании при
обработке больших
массивов данных.*

Основные принципы работы Hive



Принципы

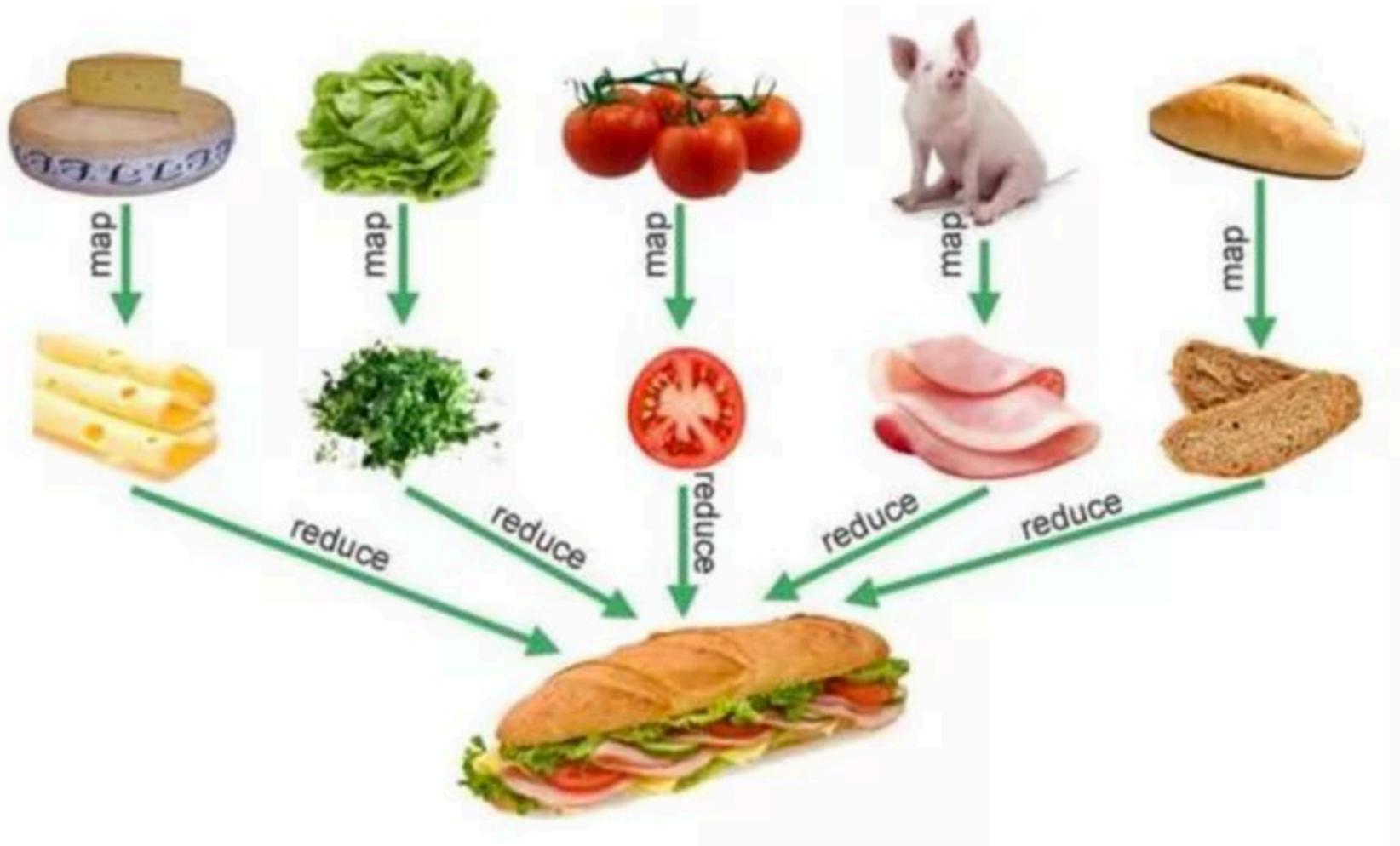
Основные принципы работы Hive



Принципы

1

Масштабируемость MapReduce



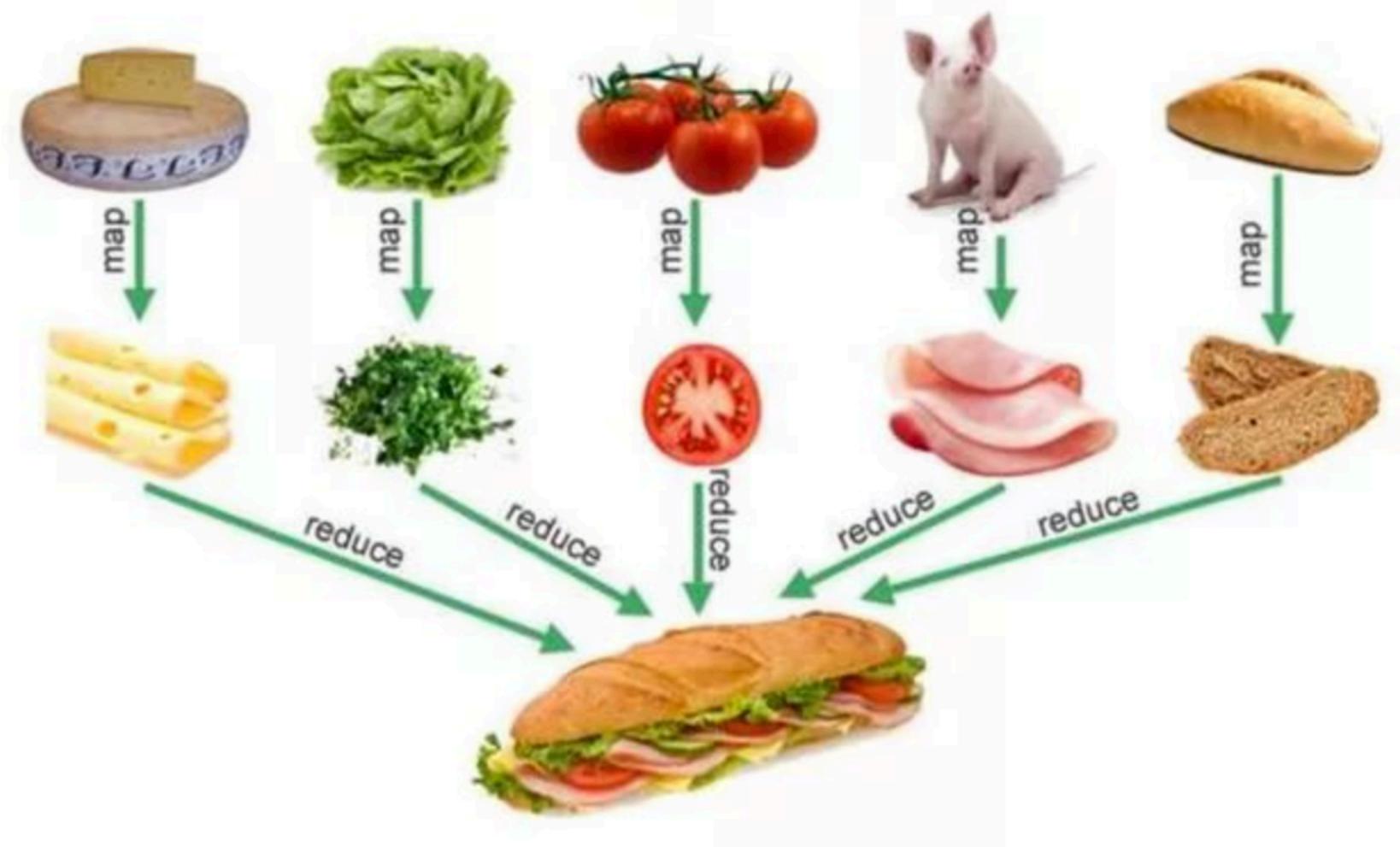
Основные принципы работы Hive



Принципы

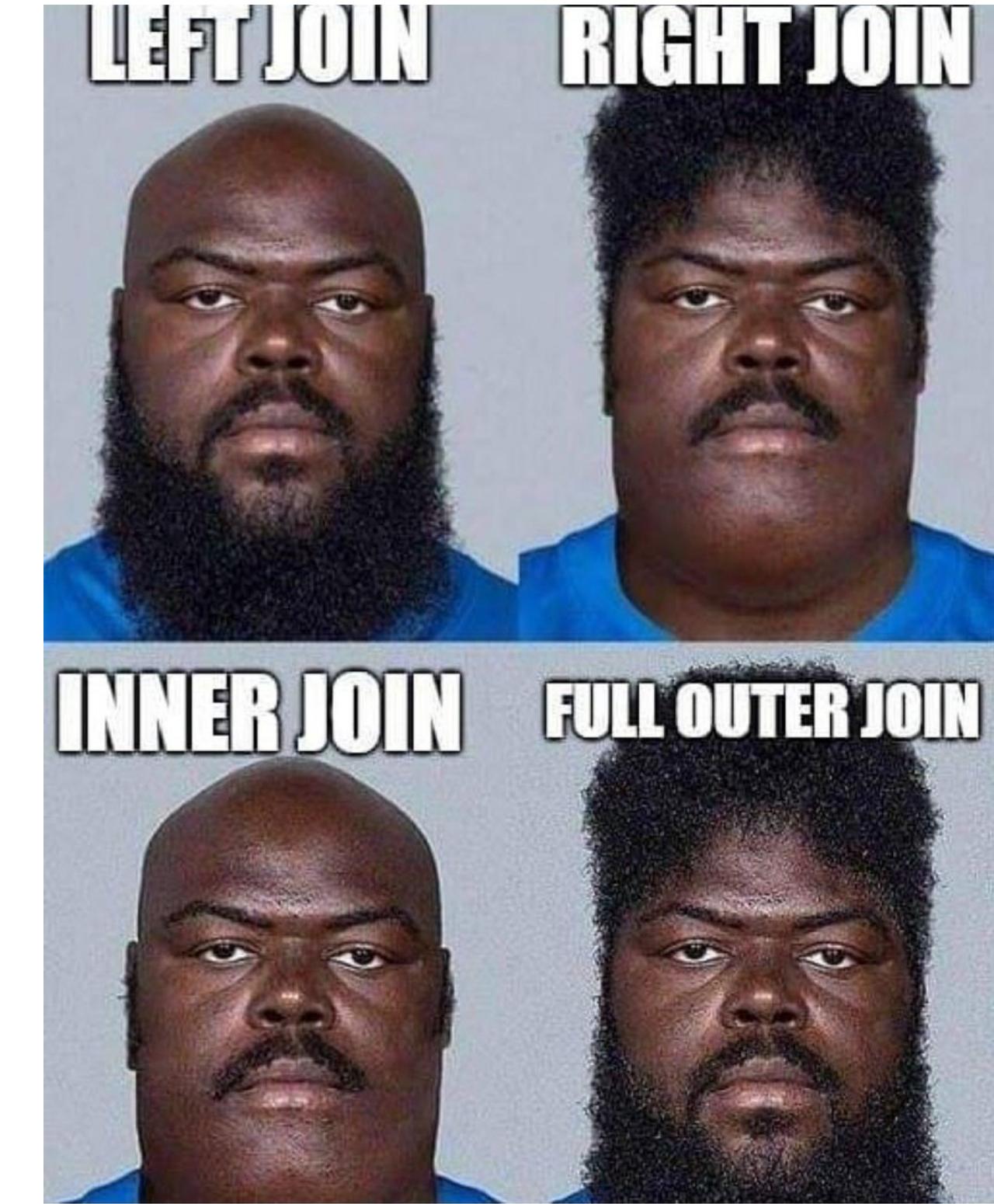
1

Масштабируемость MapReduce

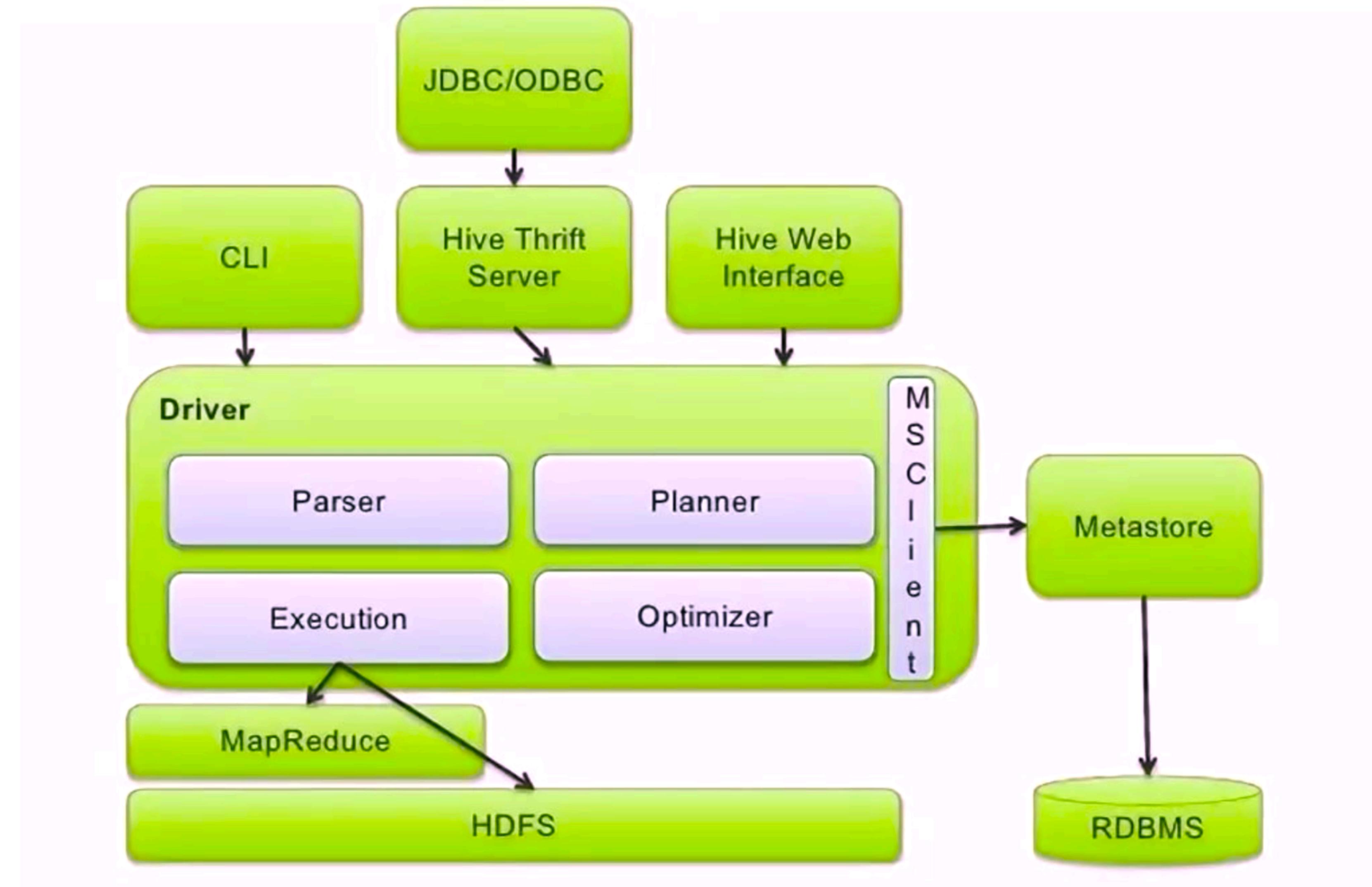


2

Удобство использования SQL для выборок из данных.



Архитектура Hive



Объекты Hive

1. База данных
2. Таблица
3. Партиция (partition)
4. Бакет (bucket)

Объекты Hive

1. База данных
2. Таблица
3. Партиция (partition)
4. Бакет (bucket)

```
CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>
```

SQL ▾

Объекты Hive

1. База данных
2. **Таблица**
3. Партиция (partition)
4. Бакет (bucket)

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name  
[(col_name data_type [COMMENT col_comment], ...)]  
[COMMENT table_comment]  
[ROW FORMAT row_format]  
[STORED AS file_format]
```

SQL ▾

Объекты Hive

1. База данных
2. Таблица
3. **Партиция (partition)**
4. Бакет (bucket)

Объекты Hive

1. База данных
2. Таблица
3. Партиция (partition)
4. **Бакет (bucket)**

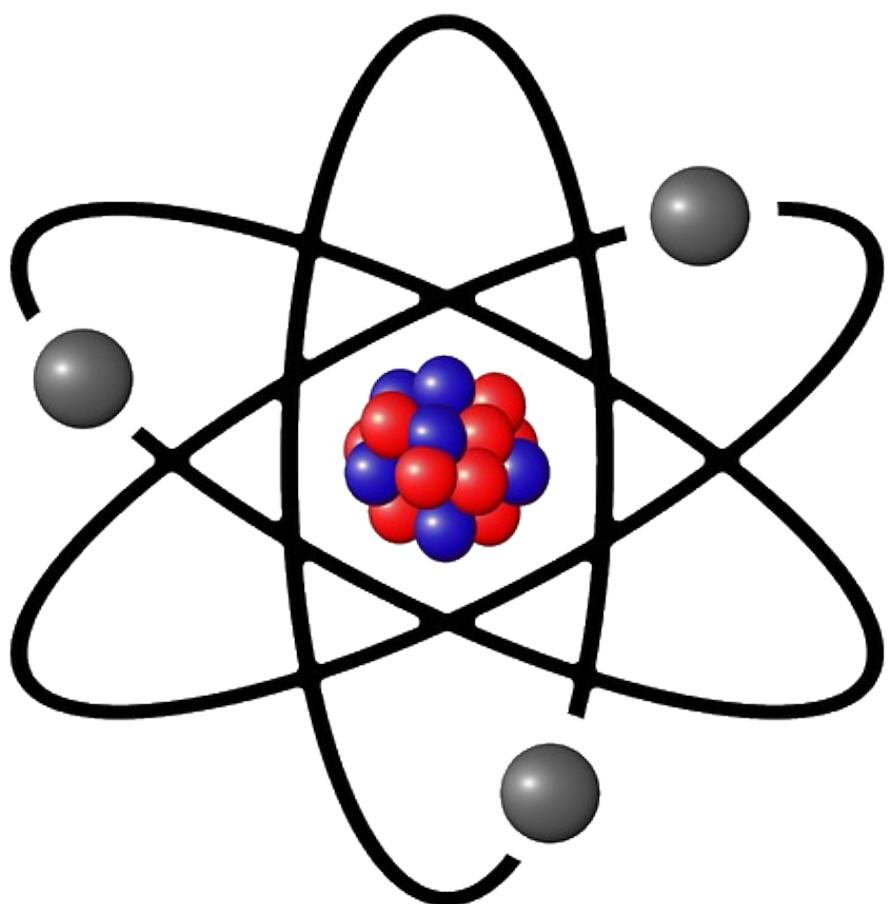
Особенности работы с Hive-запросами.

Когда стоит использовать Hive

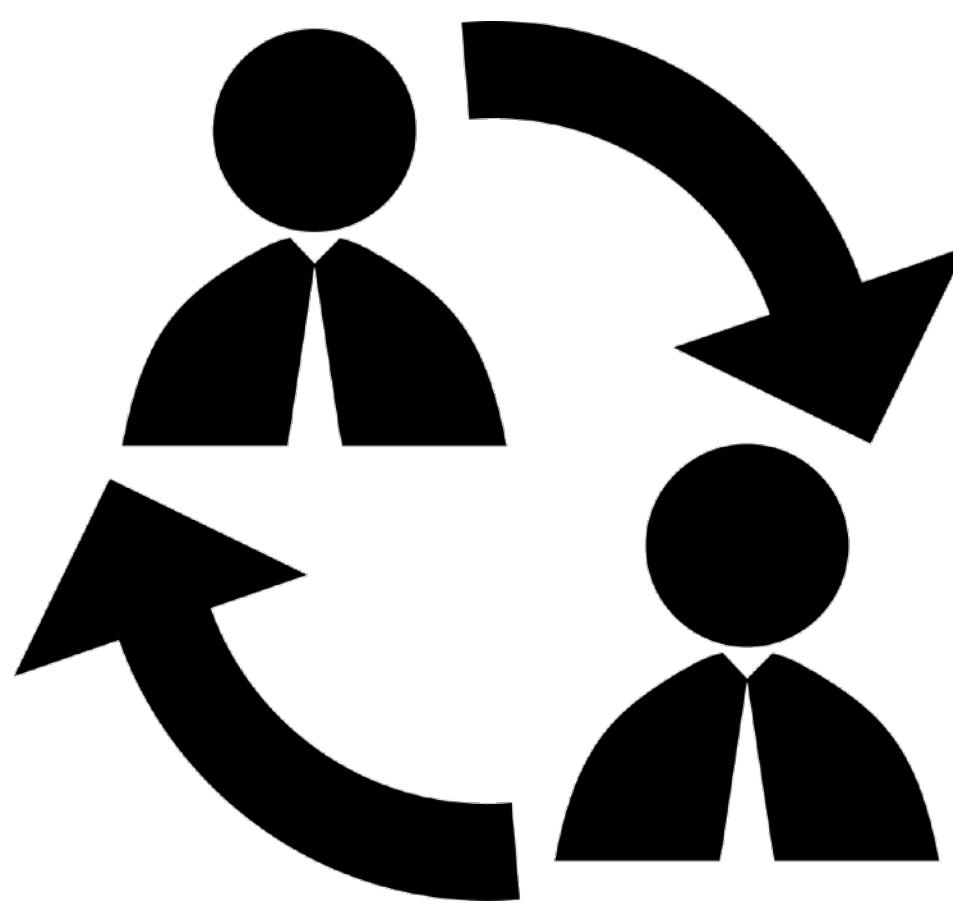
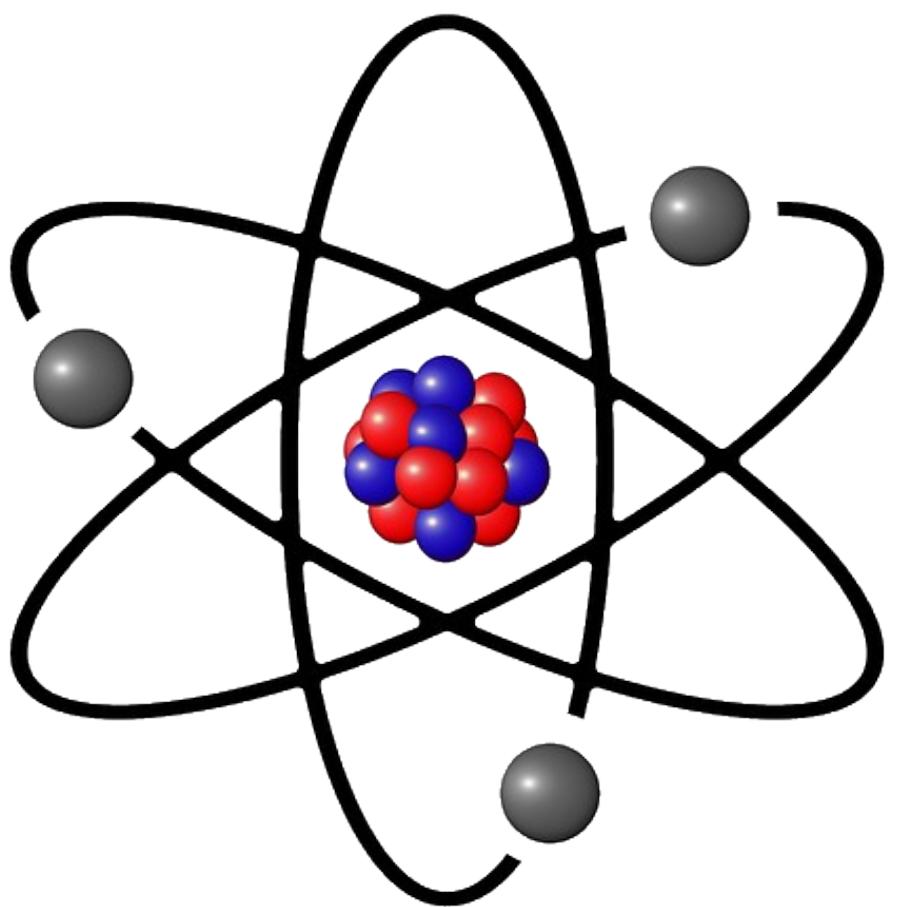
1. Данных очень много и они не умещаются на жесткий диск одной машины
2. Данные лишь добавляются и требуют редкого обновления
3. Данные структурированы
4. Для работы с данными подходят средства и шаблоны SQL
5. Время обработки запросов не критично

NoSQL поверх больших данных

NoSQL поверх больших данных



NoSQL поверх больших данных

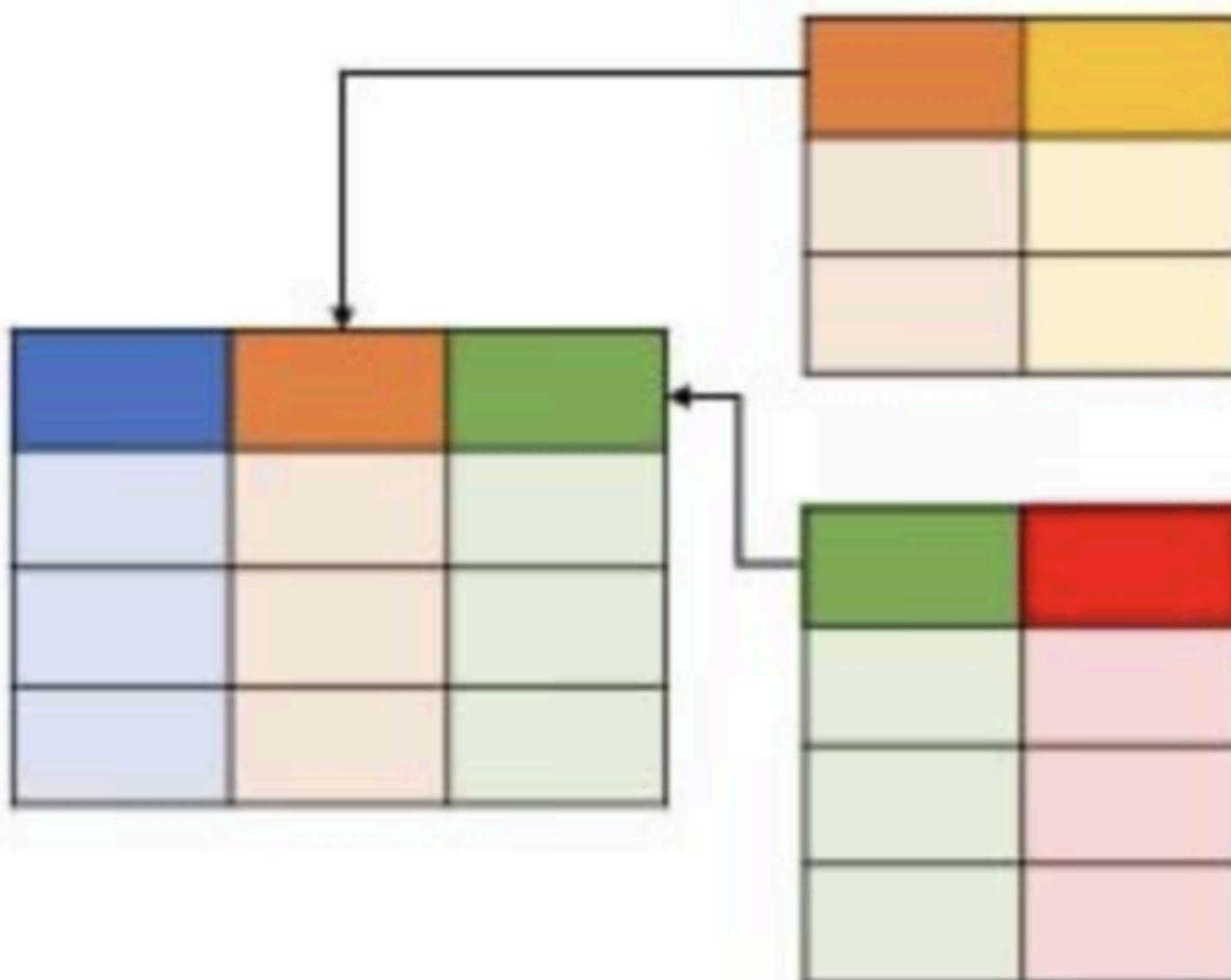


Различия SQL и NoSQL

<u>Аа</u> Свойство	≡ SQL	≡ NoSQL
<u>Структура и тип хранящихся данных</u>	Жесткие ограничения на структуру данных	Нет ограничений
<u>Запросы</u>	Запросы на языке SQL	Разные реализации для разных БД
<u>Горизонтальная Масштабируемость</u>	Сложно	Просто
<u>Надёжность</u>	Надежные решения	Не всегда является надежным хранилищем данных
<u>Поддержка</u>	Высокая популярность и долгая история, как следствие простая и легкая поддержка	Сложные пути поиска проблем
<u>Хранение и доступ к сложным структурам данных</u>	Простой доступ обеспечивается реляционностью	Сложный доступ
<u>Возможность миграции</u>	Не сложная, в связи с тем, что используется декларативный SQL язык и реляционный подход	Сложный. Каждая СУБД может иметь собственные реализации как модели данных так и языка запросов к ним

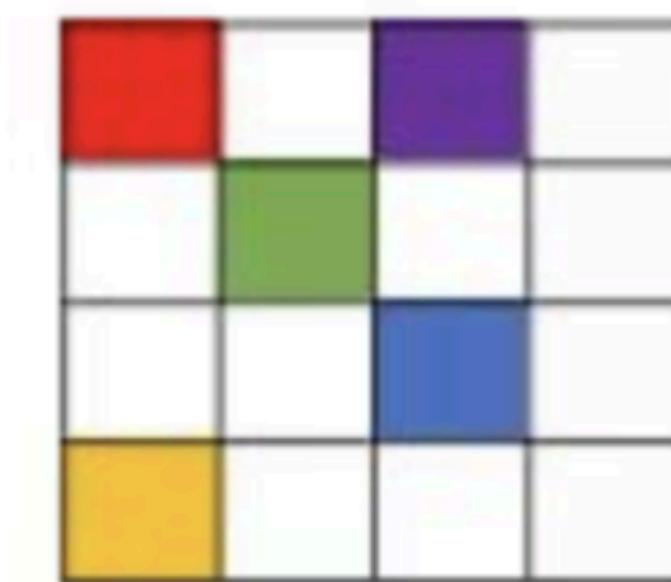
Типы NoSQL баз данных

SQL DATABASES

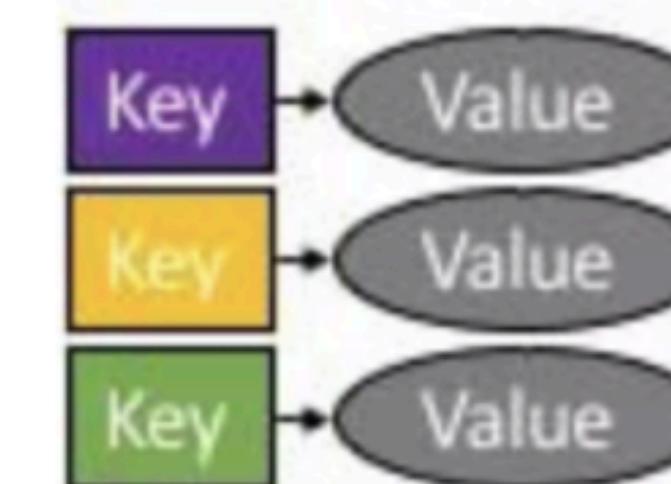


Relational

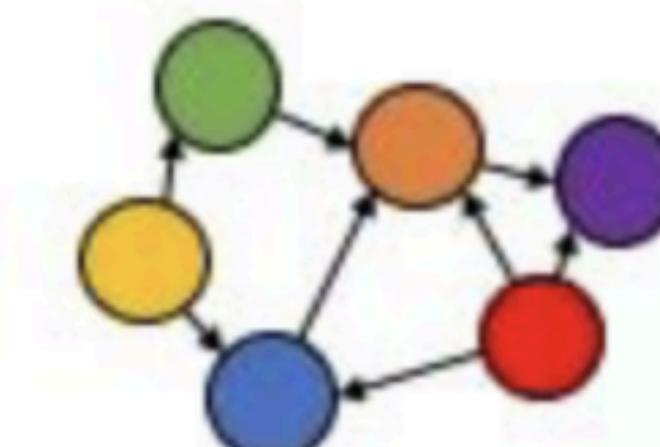
NoSQL DATABASES



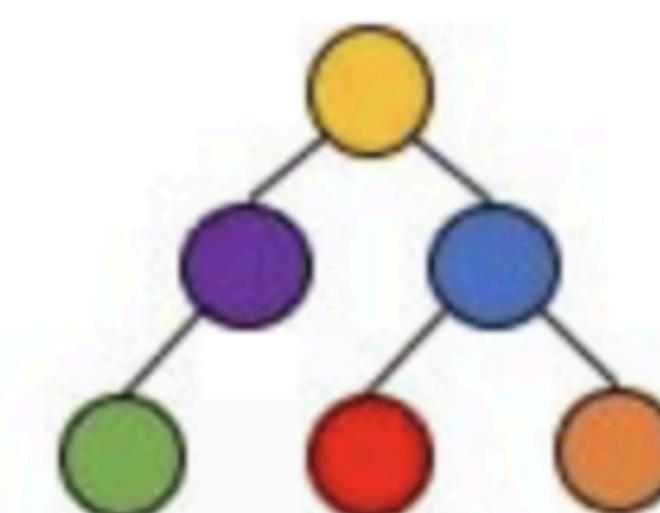
Column



Key-Value



Graph



Document

Недостатки NoSQL

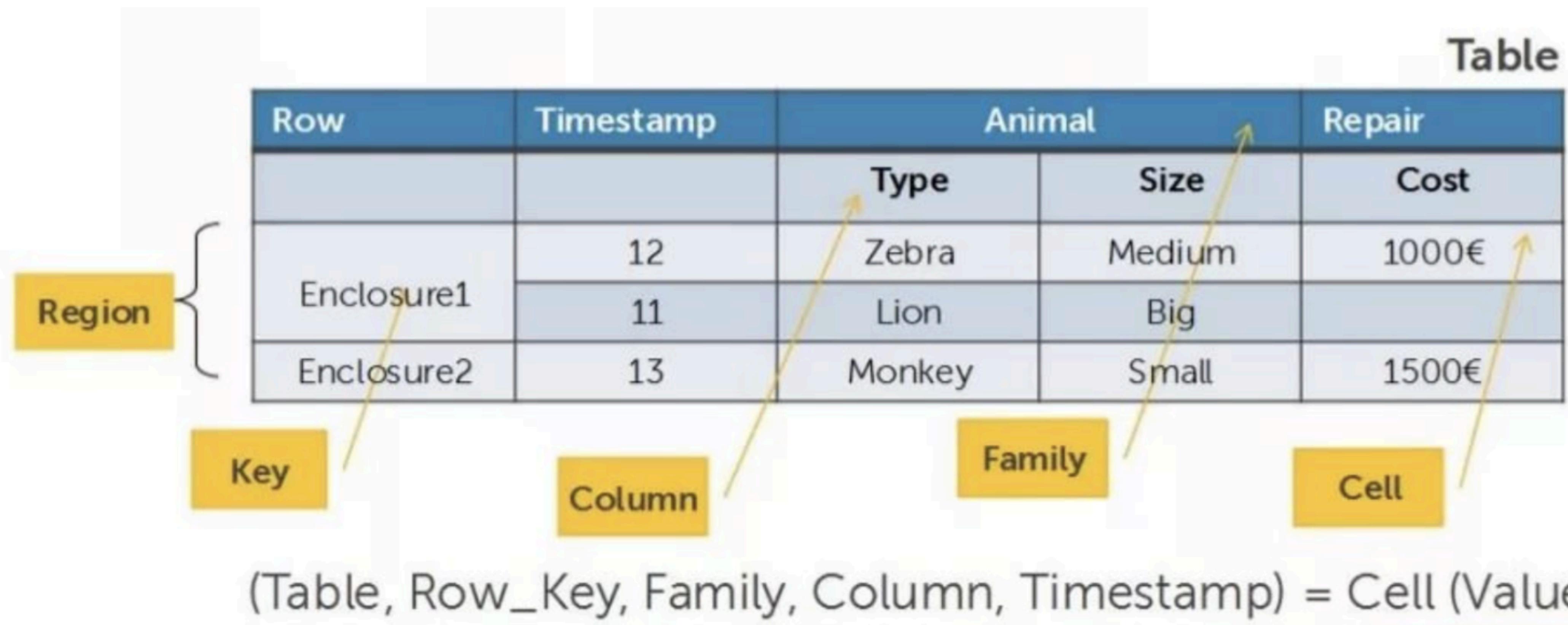
NoSQL не поддерживает:

- Транзакционность
- Реляционную аналитику (*GROUP BY, JOIN, WHERE column и т.д.*)
- доступ на основе текстовых запросов (*LIKE <text>*)

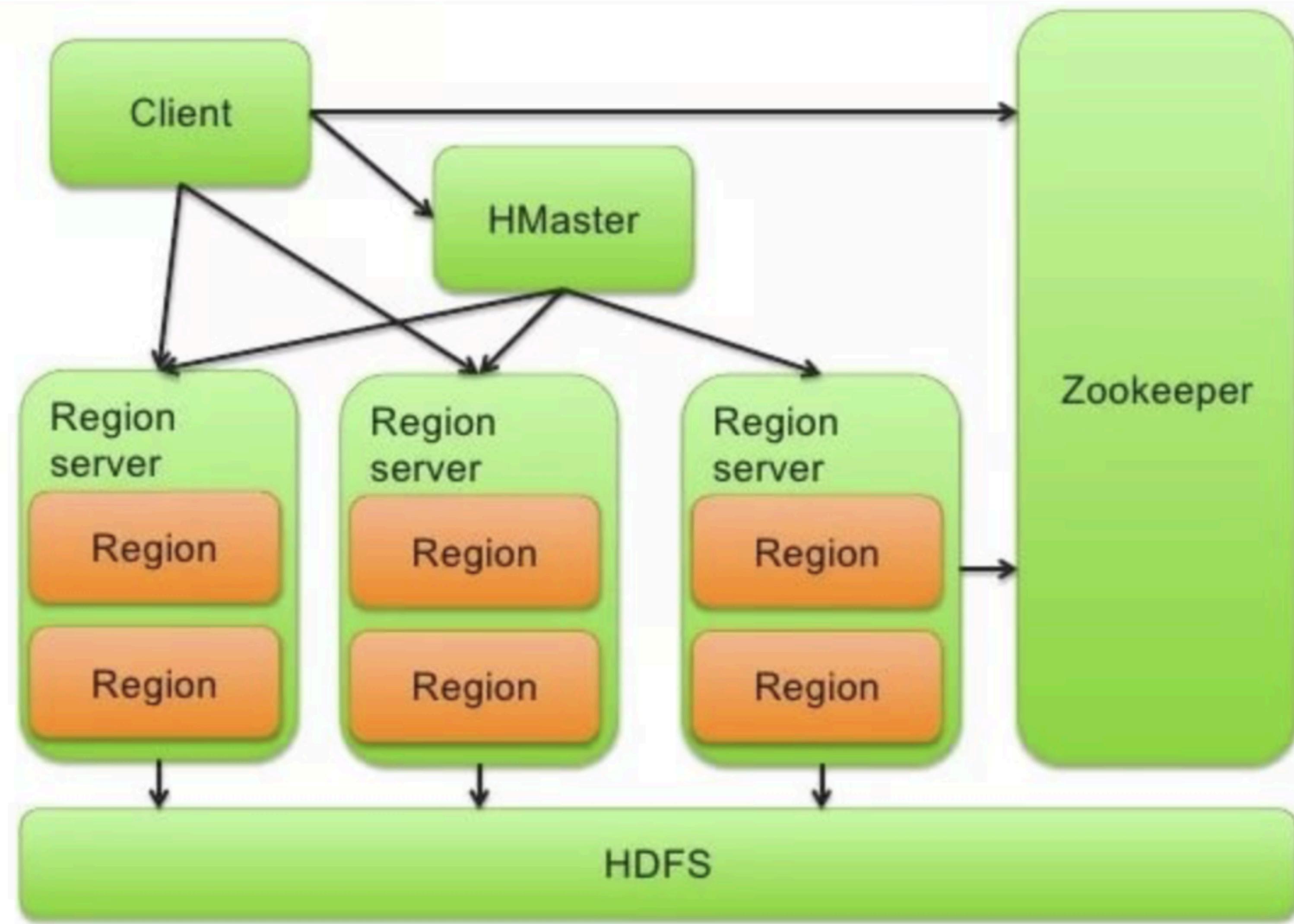
NoSQL поверх больших данных



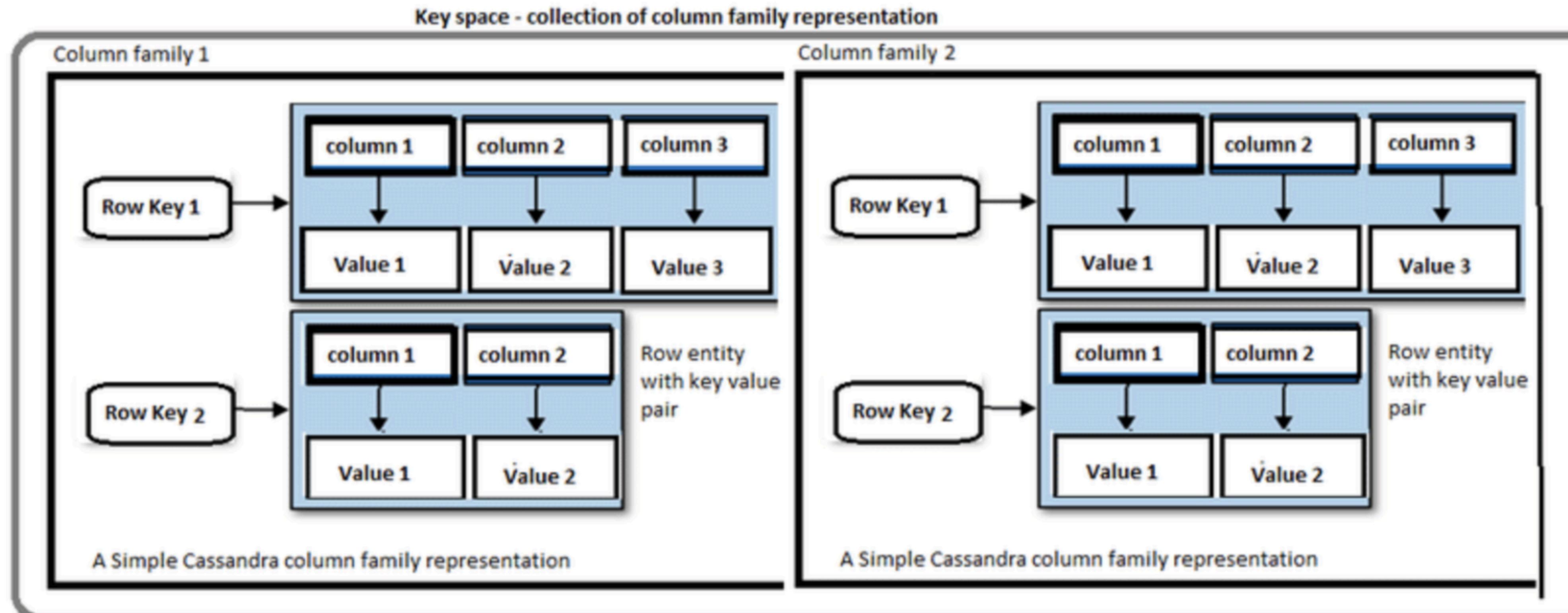
Модель данных HBase



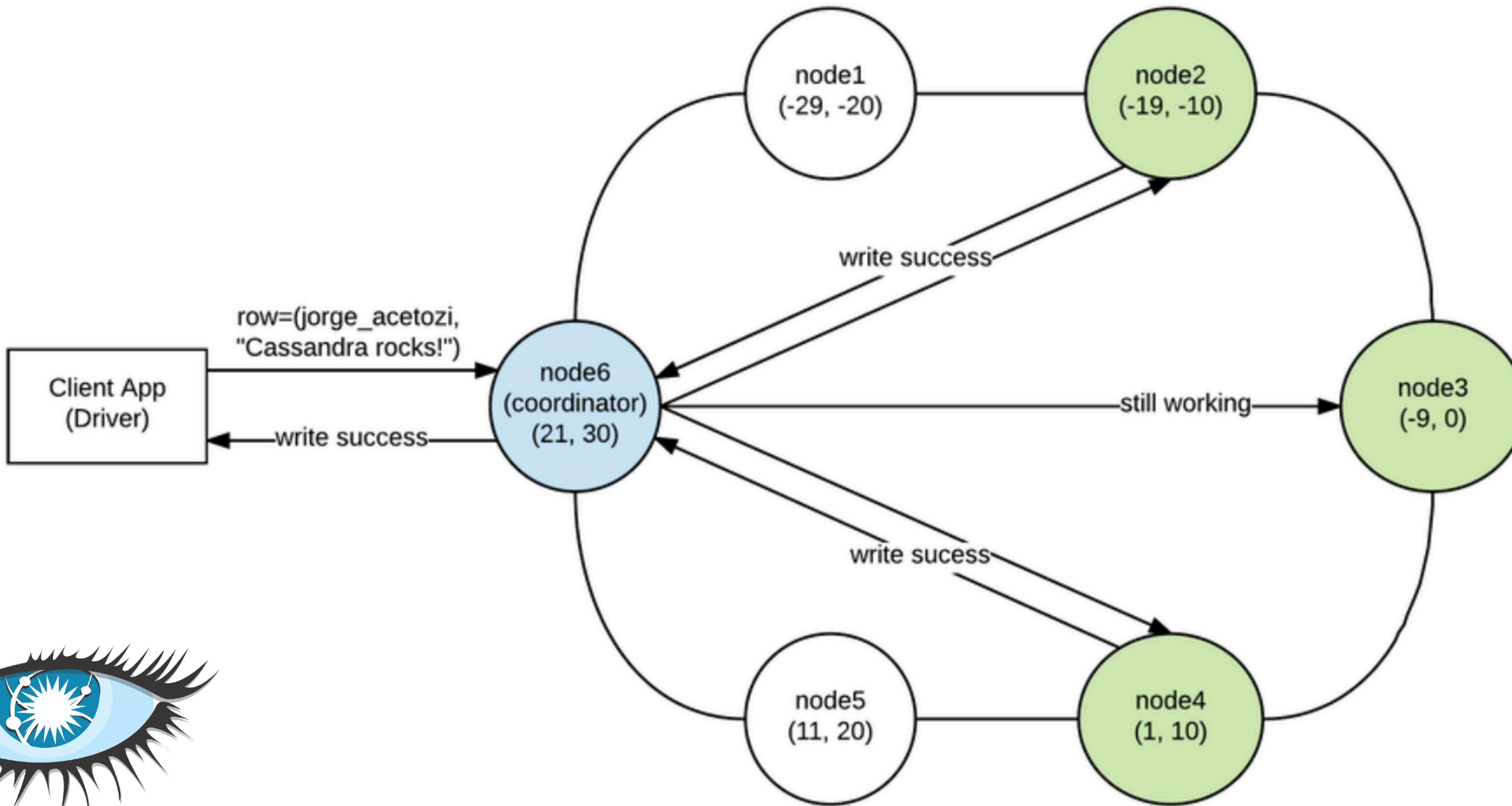
Архитектура HBase



Модель данных Apache Cassandra



Архитектура Apache Cassandra



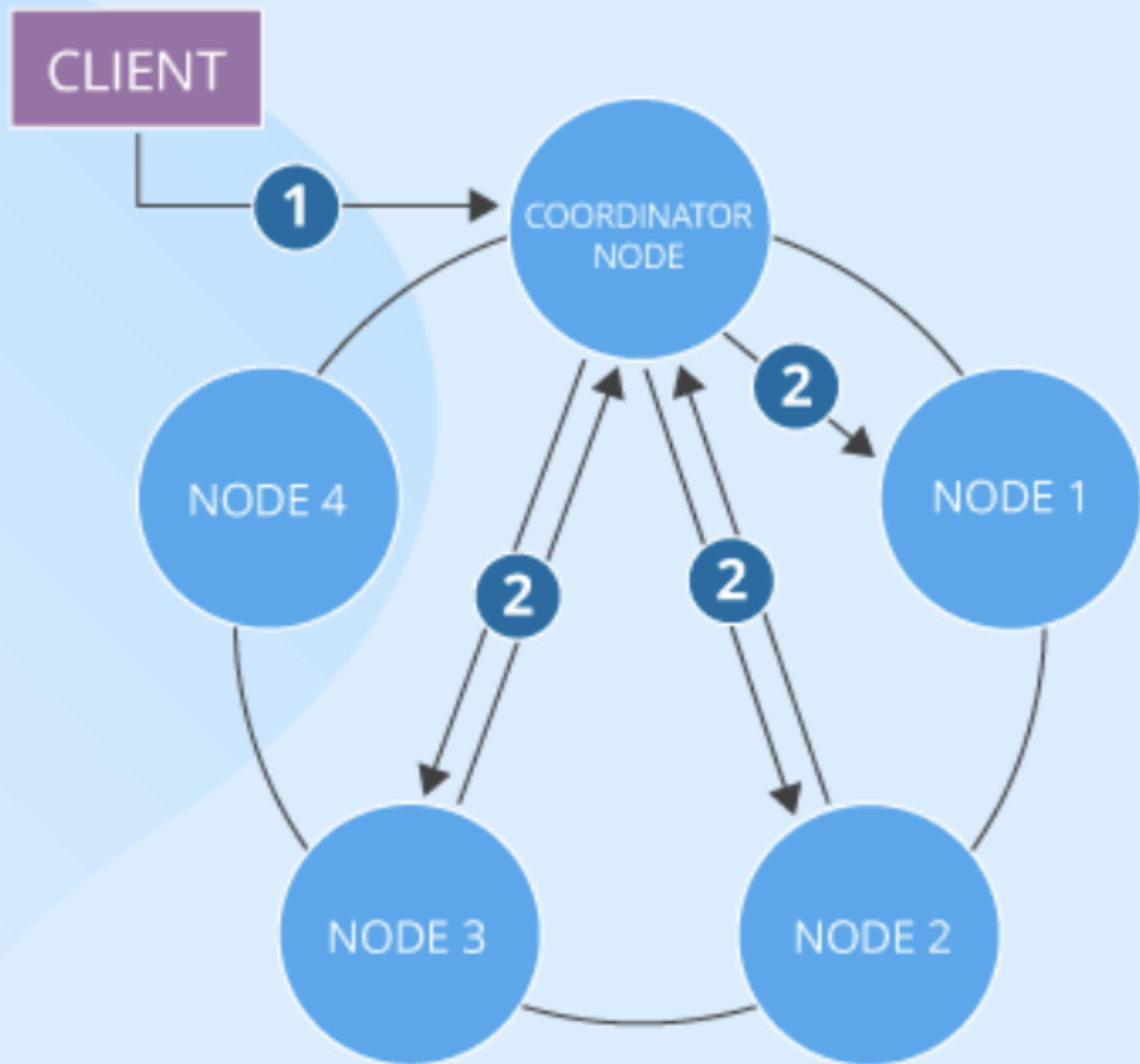
Сравнение HBase и Cassandra



<u>Единая точка отказа</u>	+ (MasterServer)	-
<u>Надежность и доступность</u>	Низкая ★	Высокая
<u>Репликация</u>	- (На стороне HDFS)	+
<u>Согласованность данных</u>	Высокая	Низкая
<u>Скорость записи</u>	Низкая	Высокая
<u>Скорость чтения</u>	Высокая	Низкая
<u>Удаление и обновление данных</u>	Нежелательно	Нежелательно



cassandra WRITE



FIRST WRITE

