

# Лекция 15

## NoSQL

# NoSQL

NoSQL (not only SQL) — обозначение широкого класса разнородных систем управления базами данных, появившихся в конце 2000-х — начале 2010-х годов и существенно отличающихся от традиционных реляционных СУБД с доступом к данным средствами языка SQL. — *определение из Wiki*

Простыми словами, NoSQL — это общее название для нереляционных баз данных.

# Типы систем

- Ключ-значение
- Семейство столбцов
- Документоориентированная СУБД
- Графовая СУБД

# Ключ — значение

- наиболее простой вариант хранилища данных, использующий ключ для доступа к значению в рамках большой хэш-таблицы.
- Хранит хэш-таблицу ключей, где каждый ключ связан с непрозрачным бинарным объектом
- Быстрая запись, но доступ только к одной ячейке
- Легко горизонтально масштабируется, идеально для приложений с большими массивами простых данных
- Примеры: Oracle NoSQL Database, Berkeley DB, MemcacheDB, Redis, Riak, Amazon DynamoDB

# Семейство столбцов

- Другие названия – столбчатая, колоночная БД
- БД, в которой данные группируются и хранятся (физически) не по строкам, а по столбцам
- Оптимизированы для быстрого извлечения столбцов данных и применяются, как правило, в аналитических приложениях

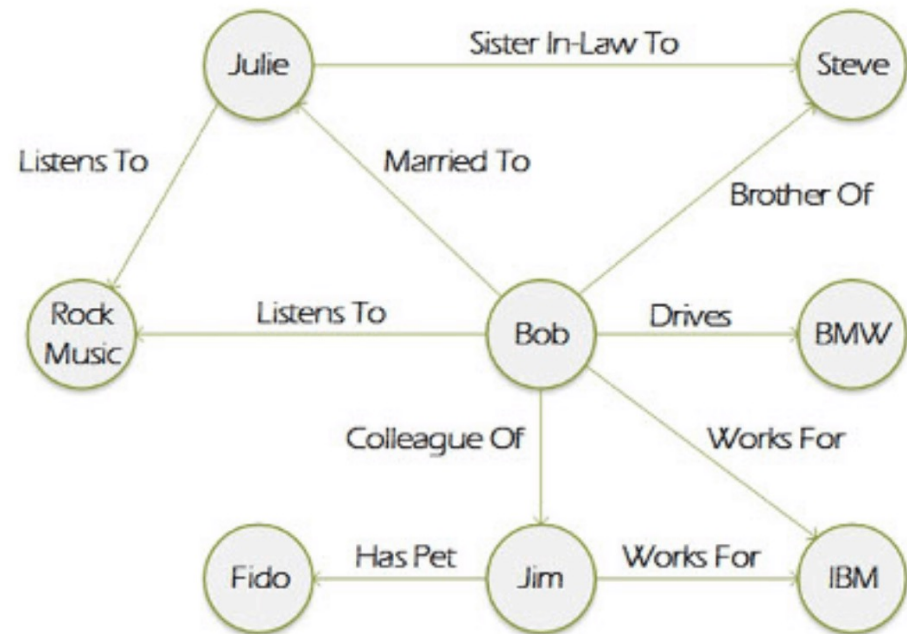
# Семейство столбцов

- Использование столбчатых таблиц баз данных для хранения оказывает большое влияние на производительность аналитических запросов, поскольку значительно снижает общие требования к операциям дискового ввода-вывода и сокращает объем данных, которые требуется загружать с диска.
- Типичным применением этого типа СУБД является веб-индексирование, а также задачи, связанные с большими данными, с пониженными требованиями к согласованности.
- Примерами СУБД данного типа являются: ScyllaDB, Hypertable, ClickHouse

# Графовая СУБД

– разновидность баз данных с реализацией сетевой модели в виде направленного графа и его обобщений.

Так как рёбра графа материализованы, то есть, являются хранимыми, обход графа не требует дополнительных вычислений (как соединение в SQL), но для нахождения начальной вершины обхода требуется наличие индексов.



# Графовая СУБД

- Графовые СУБД применяются для задач, в которых данные имеют большое количество связей, например, социальные сети, выявление мошенничества, сервисы рекомендаций. Там графовые СУБД могут существенно превосходить реляционные по производительности, а также иметь преимущества в наглядности представления и простоте внесения изменений в схему базы данных.
- С другой стороны, при незначительном количестве связей и больших объемах данных графовые БД демонстрируют значительно более низкую производительность.
- Примеры: Neo4j, OrientDB, Amazon Neptune, FlockDB, Titan.



# Документоориентированная СУБД

— СУБД, специально предназначенная для хранения иерархических структур данных (документов) и обычно реализуемая с помощью подхода NoSQL.

```
{ "_id" : 1, "item" : "abc1", qty: 300, price: 1000 }
```

```
{ "_id" : 2, "item" : "abc2", qty: 200, price: 4000 }
```

```
{ "_id" : 3, "item" : "abc3", qty: 250, price: 600 }
```

# Документоориентированная СУБД

— СУБД, специально предназначенная для хранения иерархических структур данных (документов) и обычно реализуемая с помощью подхода NoSQL.

```
{ "_id" : 1, "item" : "abc1", qty: 300, price: 1000 }
```

```
{ "_id" : 2, "item" : "abc2", qty: 200, price: 4000 }
```

```
{ "_id" : 3, "item" : "abc3", qty: 250, price: 600, discount: 10 }
```

# Документоориентированная СУБД

- Данные, представленные парами ключ-значение, сжимаются как хранилище документов схожим с хранилищем «ключ-значение» образом, с той лишь разницей, что хранимые значения (документы) имеют определённую структуру и кодировку данных. XML, JSON и BSON — некоторые из стандартных распространённых кодировок.
- Документы могут быть сгруппированы в коллекции. Их можно считать отдалённым аналогом таблиц реляционных СУБД, но коллекции могут содержать другие коллекции. Хотя документы коллекции могут быть произвольными, для более эффективного индексирования лучше объединять в коллекцию документы с похожей структурой.

# Документоориентированная СУБД

- Документарная база рассчитана на хранение отдельных элементов, не имеющих дополнительных связей между собой.
- Хорошо работает в таких примерах использования, как каталоги, пользовательские профили и системы управления контентом, где каждый документ уникален и изменяется со временем
- Примеры СУБД данного типа — MongoDB, CouchDB, Couchbase

# Документоориентированная СУБД

- Данные, представленные парами ключ-значение, сжимаются как хранилище документов схожим с хранилищем «ключ-значение» образом, с той лишь разницей, что хранимые значения (документы) имеют определённую структуру и кодировку данных. XML, JSON и BSON — некоторые из стандартных распространённых кодировок.
- Документы могут быть сгруппированы в коллекции. Их можно считать отдалённым аналогом таблиц реляционных СУБД, но коллекции могут содержать другие коллекции. Хотя документы коллекции могут быть произвольными, для более эффективного индексирования лучше объединять в коллекцию документы с похожей структурой.

# Набор свойств NoSQL

В то время как традиционные СУБД ориентируются на требования ACID к транзакционной системе, в NoSQL рассматривается набор свойств BASE:

- базовая доступность (basic availability) — каждый запрос гарантированно завершается (успешно или безуспешно)
- гибкое состояние (soft state) — состояние системы может изменяться со временем, даже без ввода новых данных, для достижения согласования данных
- согласованность в конечном счёте (eventual consistency) — данные могут быть некоторое время рассогласованы, но приходят к согласованию через некоторое время

# Теорема Брюера

Теорема CAP (известная также как теорема Брюера) — база данных может обладать не более чем двумя свойствами одновременно:

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

# Набор свойств NoSQL

Основная черта NoSQL-подхода – решение проблемы масштабируемости и доступности за счет атомарности и согласованности данных.

Таким образом, проектировщики NoSQL-систем жертвуют согласованностью данных ради достижения двух других свойств из теоремы CAP.



# Еще про NoSQL

Другими характерными чертами NoSQL-решений являются:

- Применение различных типов хранилищ.
- Возможность разработки базы данных без задания схемы.
- Линейная масштабируемость (добавление процессоров увеличивает производительность).