Севастопольский государственный университет
Кафедра «Информационные системы»

Управление данными

курс лекций

лектор:
ст. преподаватель кафедры ИС Абрамович А.Ю.

Лекция 17

NoSQL: виды, особенности и применение

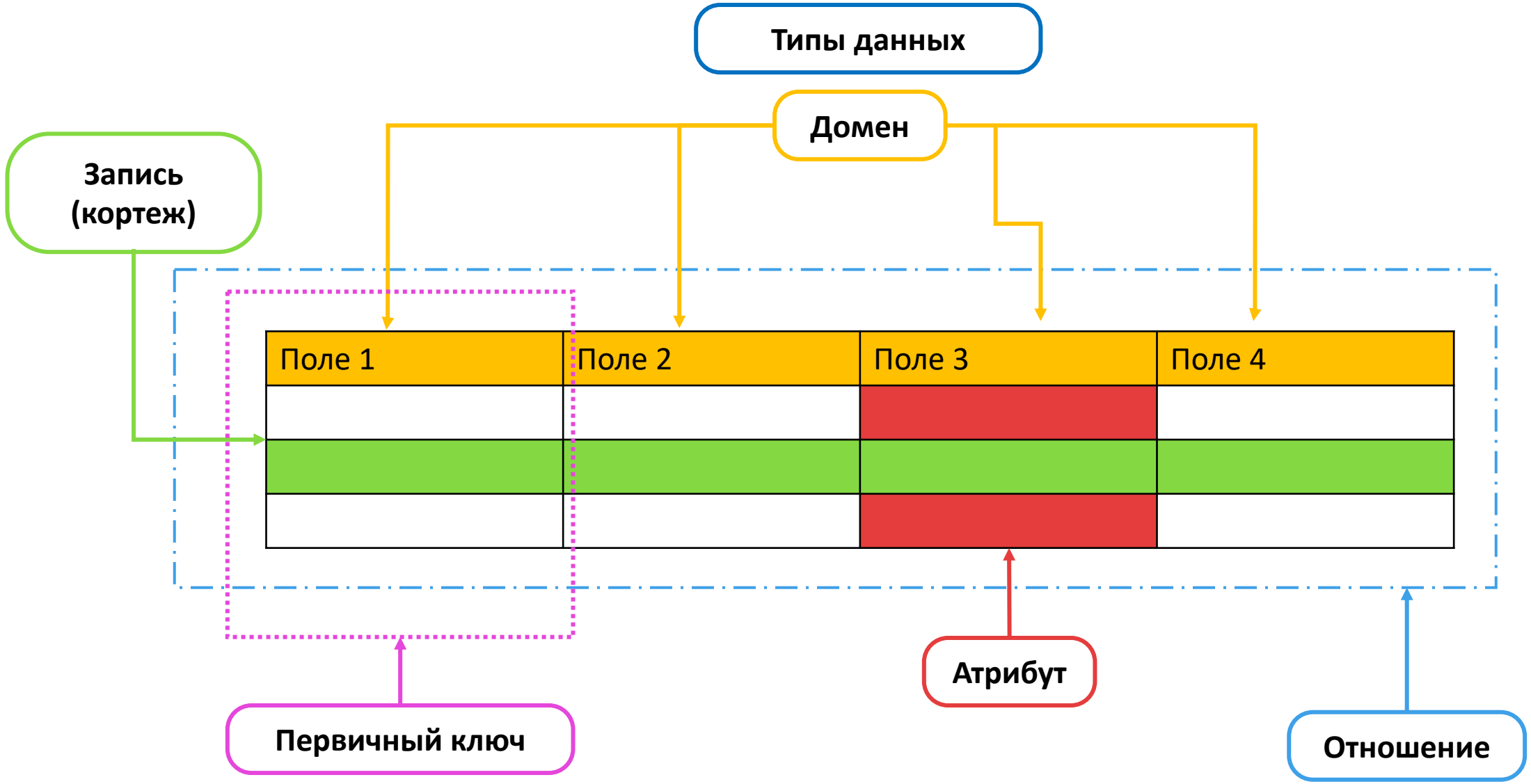
ПРОБЛЕМА БОЛЬШИХ ДАННЫХ

Big data представляет собой безмерный объем информации, который не может быть обработан стандартными инструментами и аппаратными средствами**. Основными задачами Big Data являются хранение и обработка информации гигантских объёмов данных.**

РАЗМЕР БОЛЬШИХ ДАННЫХ ОПРЕДЕЛЯЕТСЯ ОТ НЕСКОЛЬКИХ ДЕСЯТКОВ ТЕРАБАЙТ ДО 100 ЗЕТТАБАЙТ (2^{70}).

Объем хранимой информации **удваивается каждые два года**. Из всего объема существующих данных **потенциально полезны 22%,** из которых **менее 5% были подвергнуты анализу.**

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ



НЕДОСТАТКИ РЕЛЯЦИОННОЙ МОДЕЛИ

ACID свойства (атомарность, согласованность, изолированность, долговечность) **не позволяют наращивать производительность реляционных систем.**

РЕШЕНИЕ ПРОБЛЕМЫ ПРОИЗВОДИТЕЛЬНОСТИ РЕЛЯЦИОННЫХ СУБД

ИСПОЛЬЗОВАТЬ БОЛЕЕ МОЩНОЕ ОБОРУДОВАНИЕ (ВЕРТИКАЛЬНОЕ МАСШТАБИРОВАНИЕ)

ОПТИМИЗИРОВАТЬ ЗАПРОСЫ, СОЗДАТЬ ДОПОЛНИТЕЛЬНЫЕ ИНДЕКСЫ

ДЕНОРМАЛИЗАЦИЯ СХЕМЫ БД

NoSQL РЕШЕНИЯ

NewSQL РЕШЕНИЯ

NoSQL ИСТОРИЯ ПОЯВЛЕНИЯ И РАЗВИТИЯ

NoSQL — это семейство нереляционных баз данных. В них разработчики отошли от использования традиционной табличной модели представления информации.

Модель NoSQL появилась в ответ **на необходимость оперативно обрабатывать действительно огромные объёмы данных**. Поэтому NoSQL по большей части заточена под масштабирование по горизонтали и работу с недостаточно структурированными или постоянно меняющимися данными.

Идея баз данных такого типа зародилась гораздо раньше термина «большие данные», **еще в 80-е годы прошлого века**, во времена первых компьютеров (мэйнфреймов) и **использовалась для иерархических служб каталогов**. Современное понимание NoSQL-СУБД возникло **в начале 2000-х годов**, в рамках создания **параллельных распределённых систем для высокомасштабируемых интернет-приложений, таких как онлайн-поисковики.**

Вообще термин NoSQL обозначает **«не только SQL» (Not Only SQL), характеризуя ответвление от традиционного подхода к проектированию баз данных**. Изначально так называлась **опенсорсная база данных, созданная Карло Строззи**, которая хранила все **данные как ASCII-файлы**, а вместо **SQL-запросов доступа к данным использовала шелловские скрипты**

В начале 2000-х годов Google построил свою поисковую систему и приложения (GMail, Maps, Earth и прочие сервисы), **решив проблемы масштабируемости и параллельной обработки больших объёмов данных**. Так была создана **распределённые файловая и координирующая системы**, а также **колоночное хранилище** (column family store), основанное на вычислительной модели **MapReduce**.

В 2007 году другой ИТ-гигант, **Amazon.com**, публикует статьи о своей высокодоступной базе данных **Amazon DynamoDB**. Далее в эту гонку NoSQL- технологий для управления большими данными включилось множество корпораций: IBM, Netflix, eBay, Hulu, Yahoo! и другие ИТ-компаний со своими проприетарными и открытыми решениями.

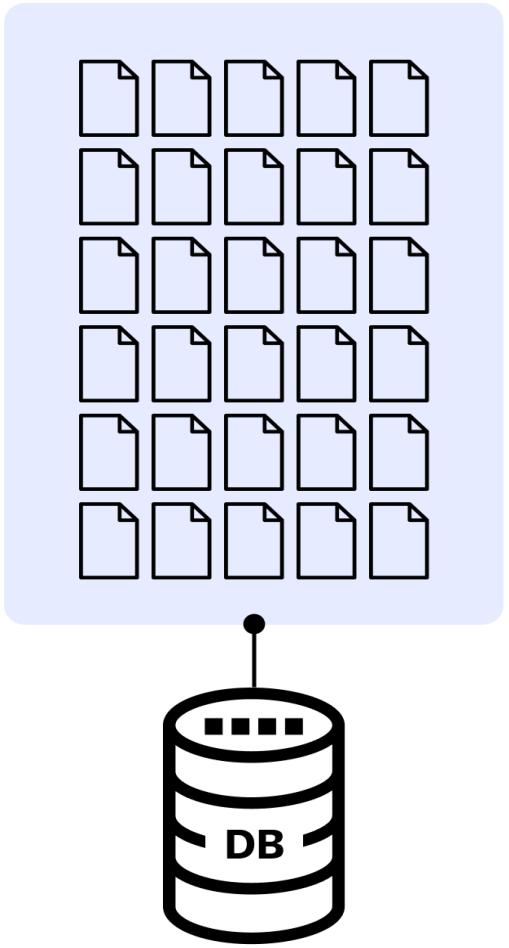
Amazon DynamoDB

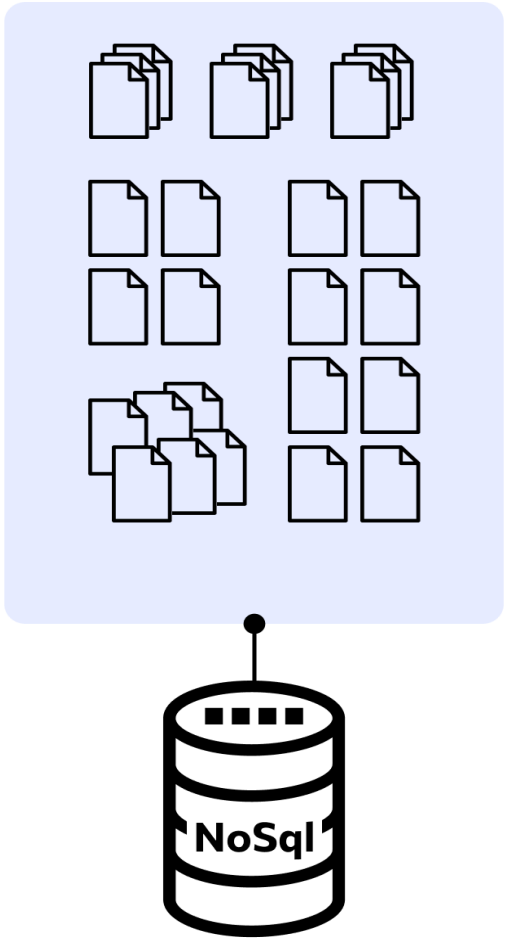
В своем смысле термин NoSQL стал применяться с 2009 года — с одноименной IT-конференции, посвященной новым опенсорсным продуктам на рынке обработки и хранения данных.

КАК РАБОТАЮТ НЕРЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

В **реляционных базах данных информация хранится в таблицах** — наборах связанных друг с другом (поэтому реляционных) записей, организованных в столбцах и строках.

В нереляционных БД строго определенной схемы взаимосвязи между данными нет. То есть информационная модель определяется СУБД «по ходу дела» в процессе работы приложения. Это **позволяет быстро адаптировать базу данных в зависимости от того, с каким типом информации в конкретный момент времени работает приложение**. Для решения аналогичной задачи с помощью реляционных БД пришлось бы подключать дополнительную базу данных.





ОСОБЕННОСТИ NOSQL

НЕСТРУКТУРИРОВАННОСТЬ.

В NoSQL-базах **структура данных не регламентирована вообще или лишь в малой степени**. Если нужно внести изменения в поля отдельного документа, для этого не потребуется декларативно менять всю структуру таблицы. При необходимости поменять модель данных потребуется просто указать изменения в коде приложения.

ИСПОЛЬЗОВАНИЕ АЛЬТЕРНАТИВ SQL.

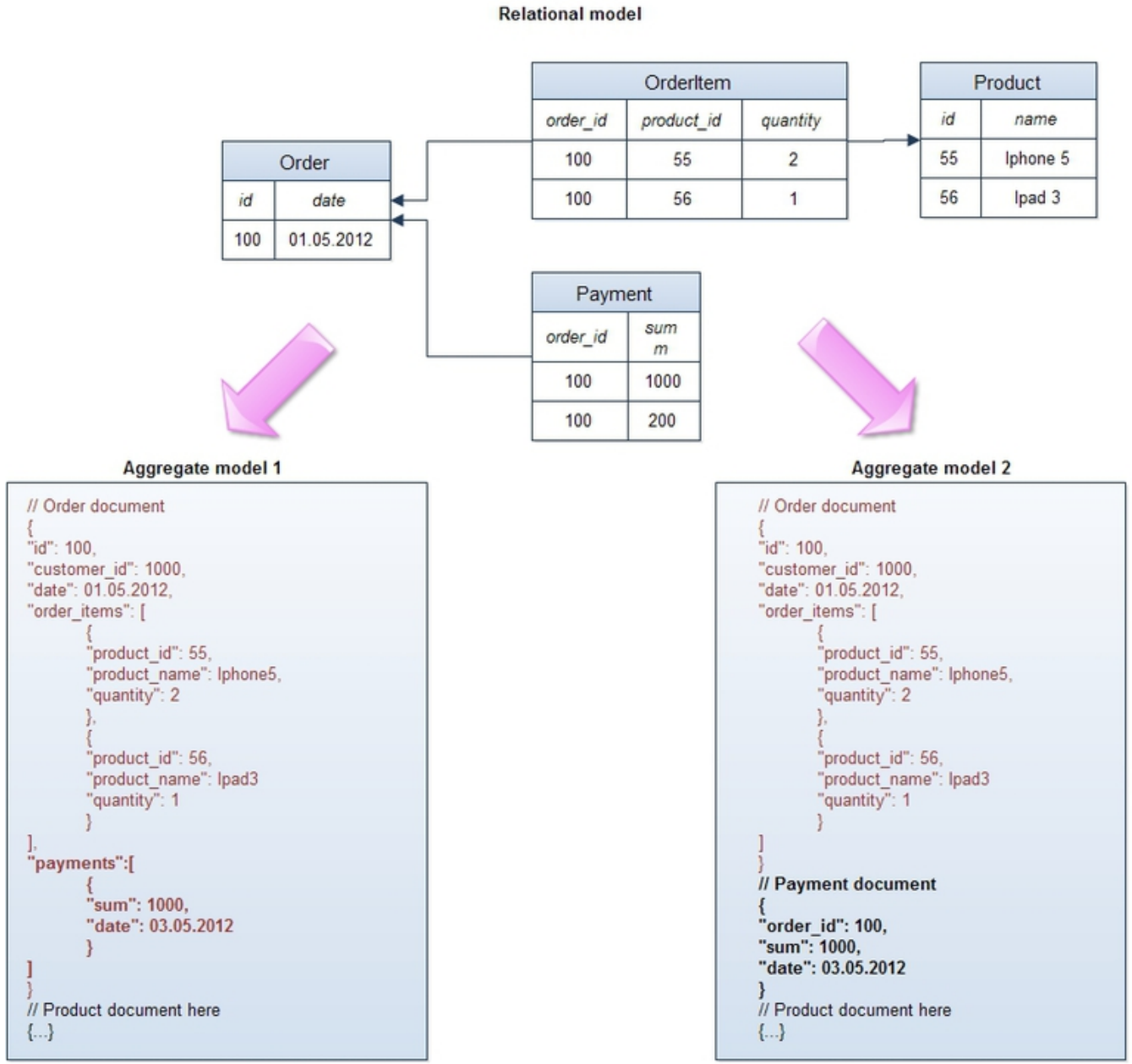
В нереляционных базах данных **не применяется именно SQL**, утвержденный Американским национальным институтом стандартов. В то же время разработчики многих NoSQL-СУБД **применяют языки управления данными, в той или иной мере похожие на него по синтаксису**.

ОТКРЫТОСТЬ.

Эта характеристика свойственна не всем, но **большинству NoSQL-базам данных**. На это повлияло и то, что их появление совпало с развитием движения за свободное (и открытое) ПО, и то, что опенсоурсные СУБД проще адаптировать под разное железо и задачи, они более доступны для изменения, проверки и т.д.

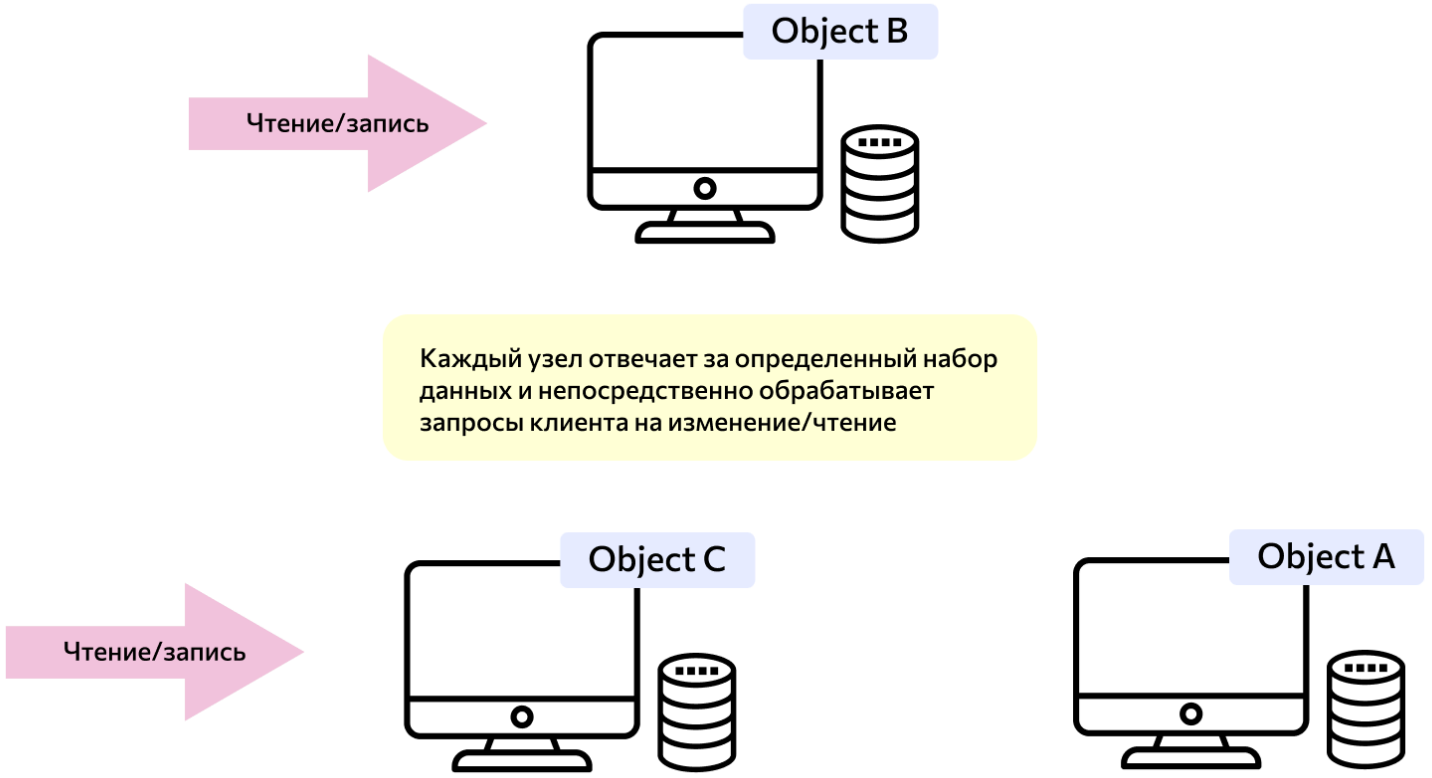
АГРЕГАЦИЯ ДАННЫХ.

В то время как реляционные БД сохраняют данные в виде таблиц, **в нереляционных они представляют собой целостные объекты**. **Агрегат** — это коллекция данных, с которой происходит взаимодействие, как с отдельной единицей. Агрегация данных **адаптируется под модель** работы конкретного приложения.



РАСПРЕДЕЛЕННОСТЬ.

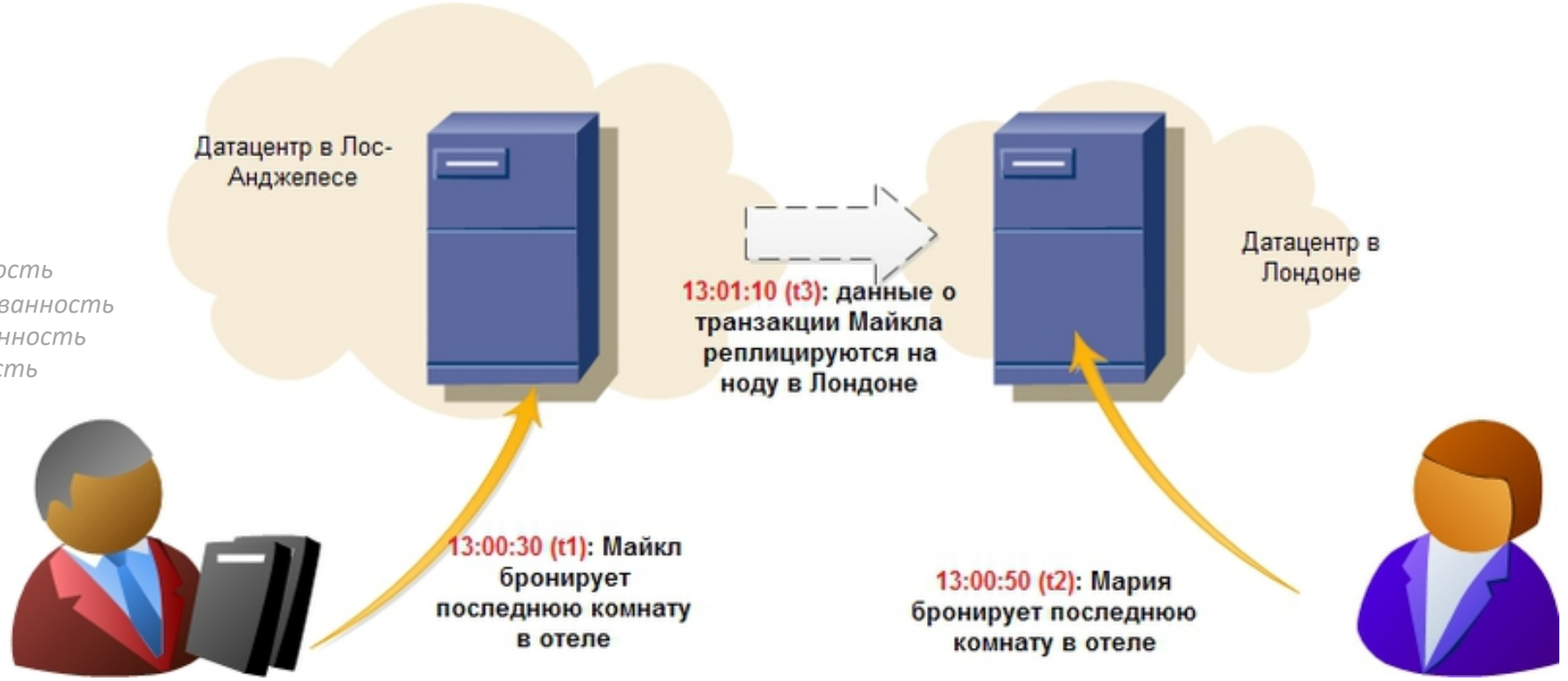
В нереляционных базах данных **реализована горизонтальная масштабируемость**. Она достигается за счет соединения быстрым подключением нескольких независимых друг от друга серверов, каждый из которых обрабатывает только часть данных, а не весь массив. Соответственно, нет необходимости наращивать мощность каждого сервера (тем более что есть физический предел) — достаточно просто добавить в систему новый.



СЛАБЫЕ ACID СВОЙСТВА.

С приходом огромных массивов информации и распределенных систем стало ясно, что обеспечить для них транзакционность набора операций с одной стороны и получить высокую доступность и быстрое время отклика с другой — невозможно. **Слабые ACID свойства не означают, что их нет вообще.** При правильном проектировании модели данных в NoSQL базе можно добиться такого же самого уровня изоляции при изменении одной записи, что и в реляционной базе данных.

Atomicity — Атомарность
Consistency — Согласованность
Isolation — Изолированность
Durability — Надежность



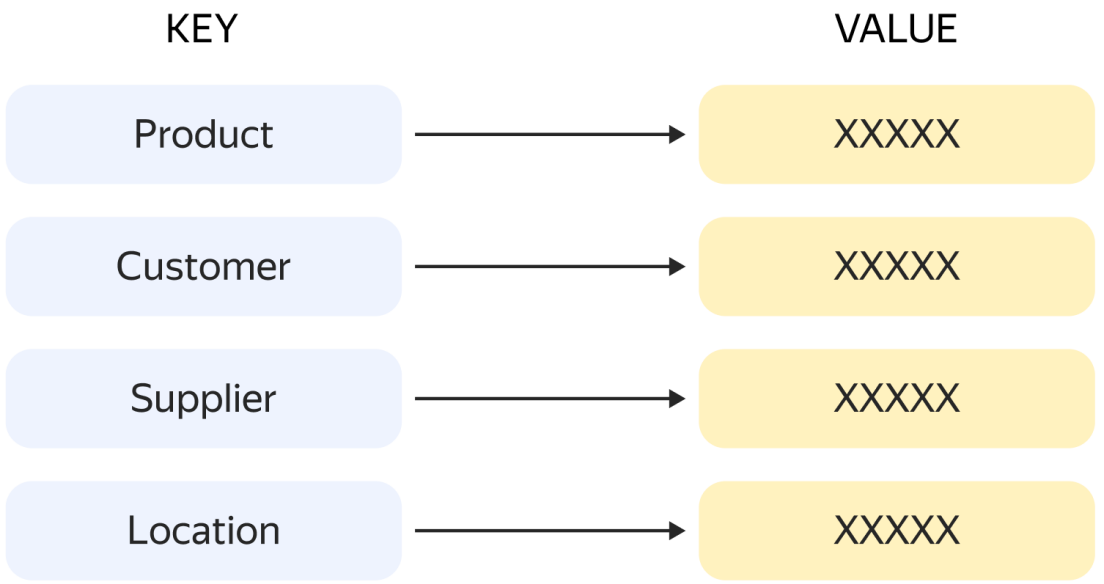
ОСНОВНЫЕ КАТЕГОРИИ НЕРЕЛЯЦИОННЫХ БД

NoSQL-СУБД различаются моделями данных, а также подходом к распределённости и репликации.

Базы данных по принципу «ключ — значение»

В этой БД **записи хранятся в парах «ключ — значение»,** где **ключ выступает уникальным идентификатором**. Ключи и значения фиксируются в виде простой или составной информации. Эти хранилища максимально быстро реагируют на запросы информации и прекрасно масштабируются.

Key-value СУБД часто используется для систем, в которых скорость является приоритетом, а данные не слишком сложные. Например, для хранения кеша данных, онлайн-списков, обработки истечения срока действия, разделения сеансов, построения рейтинга и прочих задач.



Пример key-value store БД — Redis, нереляционной СУБД с открытым исходным кодом. Ей пользуются Airbnb, Twitter и Uber. Система целиком работает в оперативной памяти, что позволяет информации считываться и записываться намного быстрее, чем даже на очень шустрые твердотельные накопители.

Базы данных по принципу «ключ — значение»

Эти БД имеют **свои столбцы и строки, но информация записывается в колонки**. Колонки **между собой не связаны, поэтому удаление или добавление новых свойств не затрагивает остальную систему**. Отсутствие заранее заданной схемы позволяет хранить в этих NoSQL-базах записи без чёткой структуры.

Колоночные (столбцовые) СУБД

В традиционной СУБД при выполнении запроса сканируется вся таблица, а информация из всей строки извлекается целиком. **В колоночных БД выгружаются только необходимые значения, поскольку поиск ведётся по отдельным столбцам.** Такой подход колоночных NoSQL баз к хранению информации позволяет быстро получать данные из больших таблиц для анализа. А возможность сильного сжатия данных экономит много места на диске.

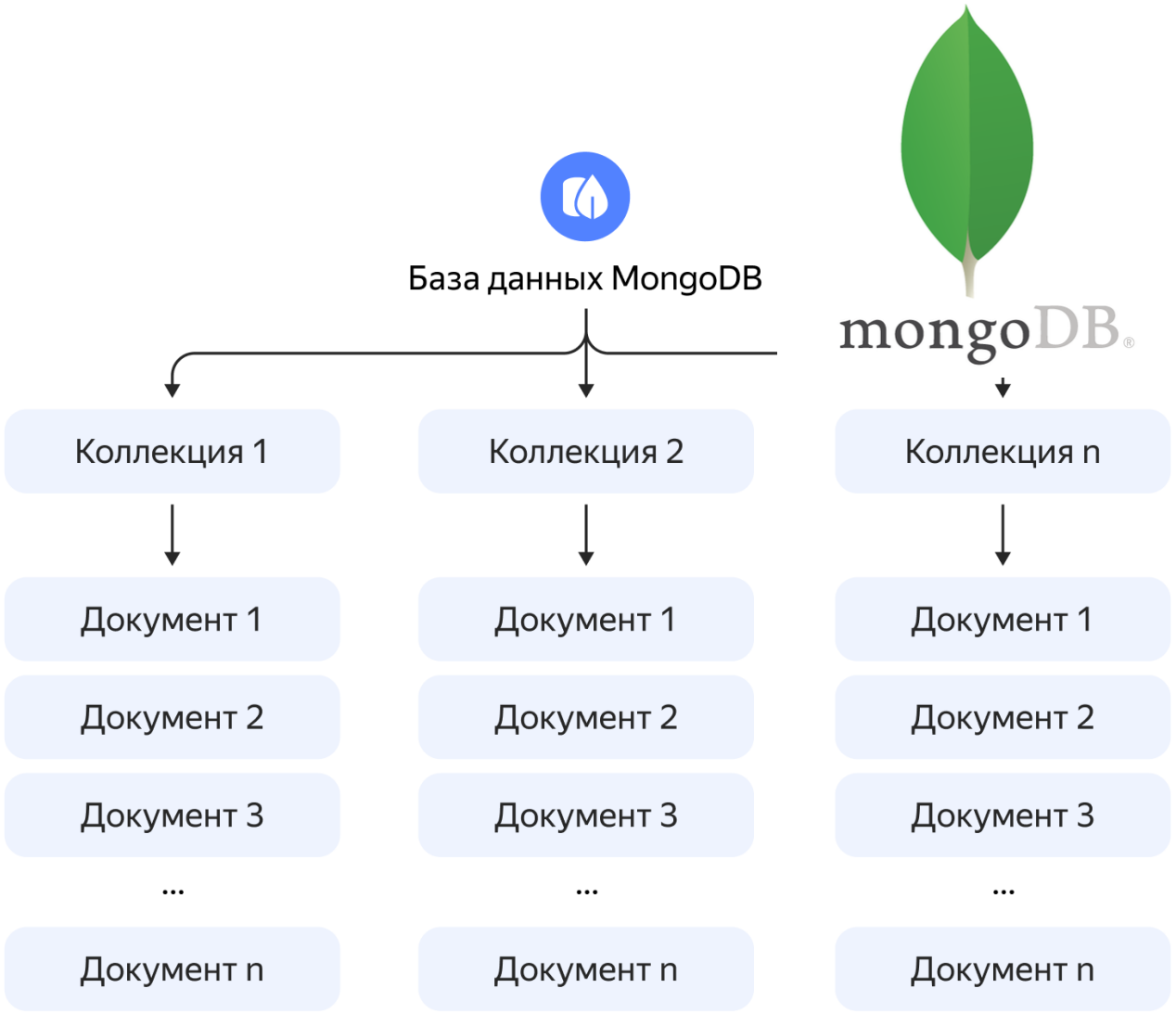
В этой категории существуют базы **ClickHouse, Vertica, Cassandra и другие**. Netflix, например, использует в том числе хранилище Cassandra: база легко масштабируется, и у неё нет единых точек отказа.

Документоориентированные базы данных (Document-oriented store)

В БД этого типа **данные записываются в документы** и **хранятся в формате, подобном JSON**. Таким хранилищам свойственны **иерархичность** (документы складываются в коллекции, а коллекции группируются логически) и **гибкость** (значения, свойства и их структура может меняться в процессе разработки).

Document-oriented-модель хороша в проектах, где нужно обрабатывать **большой объём данных без четкой структуры**, а также для работы со **множеством уникальных документов**, которые со временем требуют изменений.

Классический пример такой нереляционной СУБД — **MongoDB.**

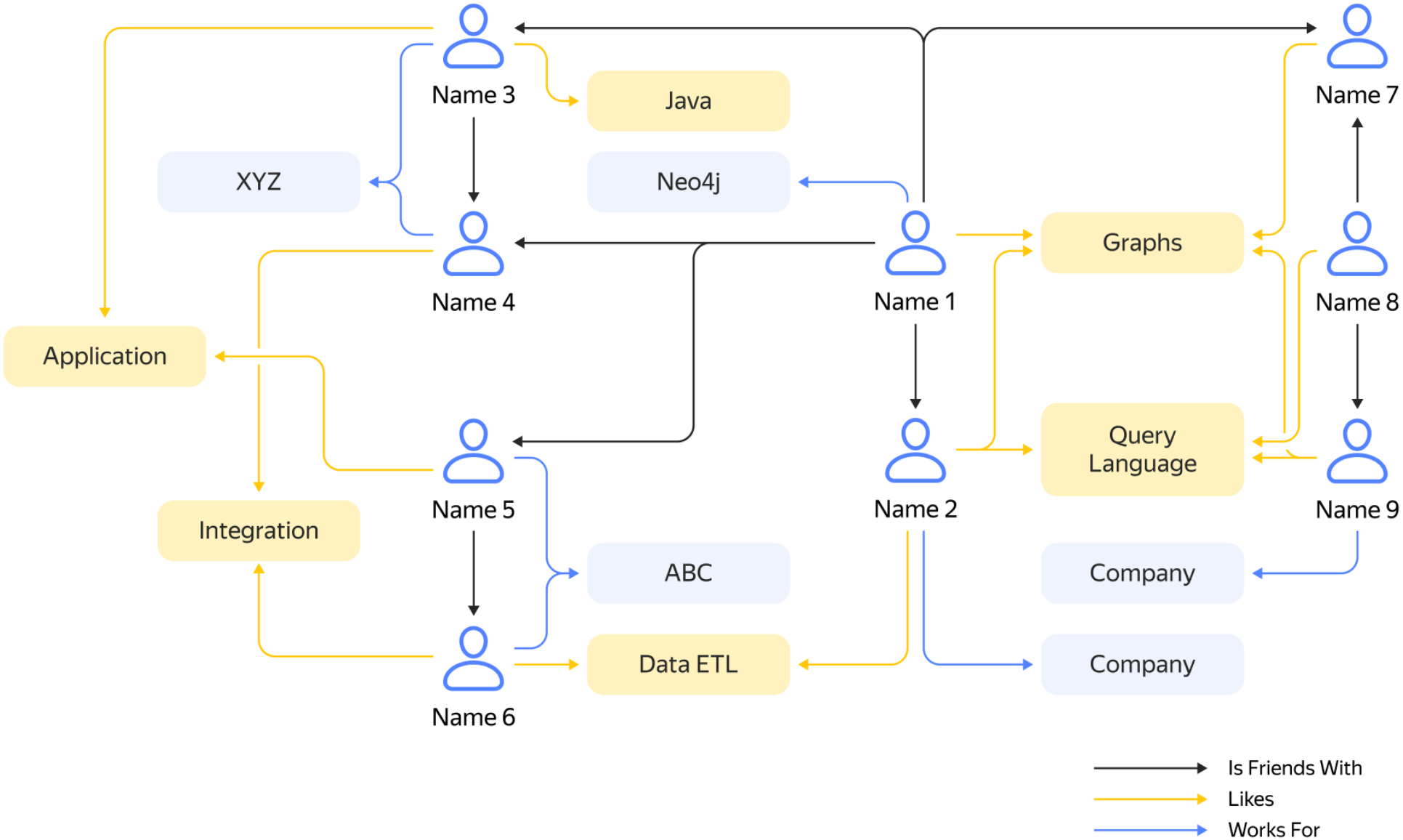


Графовые базы данных (graph store)

Элементы базы данных хранятся в узлах (вершинах), между узлами существуют ребра, которые определяют отношения элементов друг к другу. У ребра есть начальный и конечный узел, направление и тип (связи действия, права владения, «родитель — ребёнок» и пр.). **Главная особенность графовых БД — хранение не только сущностей данных, но и взаимосвязей**, тогда как в реляционных БД соединения между элементами требует дополнительных вычислений.

Благодаря такой модели данных graph store NoSQL используются для выполнения задач, ориентированных на связи: **для алгоритмов рекомендаций контента, социальных сетей, обнаружения случаев сетевого мошенничества.**

Для доступа к таким БД необходим язык запросов, но, поскольку общепринятых стандартов у NoSQL нет, для разных типов графовых баз данных понадобится индивидуальный подход. К графовым относятся базы данных **Neo4j, OrientDB**.


Name 3
Java
Name 7
XYZ
Neo4j
Graphs
Name 4
Name 1
Name 8
Application
Query
Language
Name 5
Name 2
Name 9
Integration
ABC
Company
Data ETL
Company
Name 6
Is Friends With
Likes
Works For

ПРЕИМУЩЕСТВА NoSQL

ГОРИЗОНТАЛЬНАЯ МАСШТАБИРУЕМОСТЬ — для увеличения производительности **достаточно добавить новый сервер в систему**, а не наращивать мощности уже имеющихся.

ВЫСОКАЯ УСТОЙЧИВОСТЬ — так как NoSQL-БД размещаются на независимых серверах, **выход одного из них из строя не обрушит всю базу данных** и, следовательно, не приведет к полному отказу приложения.

ПРОИЗВОДИТЕЛЬНОСТЬ — с одной стороны, **каждый сервер обрабатывает только свои запросы**, не растрачивая свои мощности на все; с другой — **информационные модели таких СУБД адаптируются под специфику** каждого приложения.

ГИБКОСТЬ — нереляционные БД **могут работать с неструктурированными данными и различными моделями** представления информации.

ШИРОКАЯ ПРИМЕНИМОСТЬ — горизонтальная масштабируемость и производительность позволяют **применять нереляционные БД в обработке больших данных, онлайн-играх, интернете вещей, электронной коммерции, научной деятельности, издательском бизнесе и т.д.**

НЕДОСТАТКИ NoSQL

ОГРАНИЧЕННОСТЬ ЯЗЫКА —в работе с нереляционными базами данных часто **приходится использовать сторонние инструменты для трансляции стандартных SQL-команд**.

НЕДОСТАТОЧНАЯ НАДЕЖНОСТЬ ТРАНЗАКЦИЙ — из-за того, что NoSQL-БД заточены под высокую производительность и масштабируемость, в них **страдает согласованность данных**, критически важная для таких сфер, как денежные переводы.