

Проектирование баз данных, ч.2

ФИО преподавателя: Смирнов Михаил Вячеславович
e-mail: smirnovmgupi@gmail.com

Лекция 12

BigData и noSQL модели данных

Каким образом можно сохранить big data

Этот способ регламентируется применением NoSQL структуры хранения данных.

NoSQL (*not only SQL, не только SQL*),

термин, обозначающий ряд подходов, направленных на реализацию хранилищ баз данных, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL.

В NoSQL вместо традиционного реляционного ACID используется 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.

Каким образом можно сохранить big data

В NoSQL вместо традиционного реляционного ACID используется 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.

Сравнение SQL и noSQL баз данных.

	Реляционные базы данных	Базы данных NoSQL
Подходящие рабочие нагрузки	Реляционные БД предназначены для транзакционных и строго непротиворечивых приложений обработки транзакций в режиме реального времени (OLTP) и хорошо подходят для аналитической обработки в режиме реального времени (OLAP).	Базы данных NoSQL (на основе пар «ключ-значение», документные, графовые и работающие в памяти) ориентированы на OLTP для целого ряда шаблонов доступа к данным, в том числе для приложений с низкой задержкой. Поисковые БД NoSQL предназначены для аналитики частично структурированных данных.
Модель данных	Реляционная модель нормализует данные и преобразует их в таблицы, состоящие из строк и столбцов. Схема жестко задает таблицы, строки, столбцы, индексы, отношения между таблицами и прочие элементы базы данных. Такая БД обеспечивает целостность ссылочных данных в отношениях между таблицами.	В базах данных NoSQL применяются различные модели данных, в том числе документные, графовые, поисковые, с использованием пар «ключ-значение» и хранением данных в памяти.

Сравнение SQL и noSQL баз данных.

	Реляционные базы данных	Базы данных NoSQL
Свойства ACID	Реляционные базы данных обеспечивают набор свойств ACID: атомарность, непротиворечивость, изолированность, надежность.	Базы данных NoSQL зачастую предлагают компромисс, смягчая жесткие требования свойств ACID ради более гибкой модели данных, которая допускает горизонтальное масштабирование.
Производительность	Производительность главным образом зависит от дисковой подсистемы. Для обеспечения максимальной производительности часто требуется оптимизация запросов, индексов и структуры таблицы.	Производительность обычно зависит от размера кластера базового аппаратного обеспечения, задержки сети и вызывающего приложения.

Сравнение SQL и noSQL баз данных.

	Реляционные базы данных	Базы данных NoSQL
Масштабирование	Реляционные базы данных обычно масштабируются путем увеличения вычислительных возможностей аппаратного обеспечения или добавления отдельных копий для рабочих нагрузок чтения.	Базы данных NoSQL обычно поддерживают высокую разделимость благодаря шаблонам доступа на основе пар «ключ-значение» с возможностью масштабирования на основе распределенной архитектуры.
API	Запросы на запись и извлечение данных составляются на языке SQL. Эти запросы анализирует и выполняет реляционная база данных.	Благодаря использованию ключей секций приложения могут вести поиск по парам «ключ-значение», наборам столбцов или частично структурированным документам, содержащим серийные объекты и атрибуты приложений. Многие приложения, работающие с BigData имеют API, ориентированный на работу с JSON

Сравнение терминологии SQL и noSQL баз данных.

SQL	MongoDB	Cassandra
Таблица	Коллекция	Таблица
Строка	Документ	Ряд
Столбец	Поле	Столбец
Первичный ключ	ObjectID	Первичный ключ
Индекс	Индекс	Индекс
Представление	Представление	Материализованное представление
Вложенная таблица или объект	Встроенный документ	Карта
Массив	Массив	Список

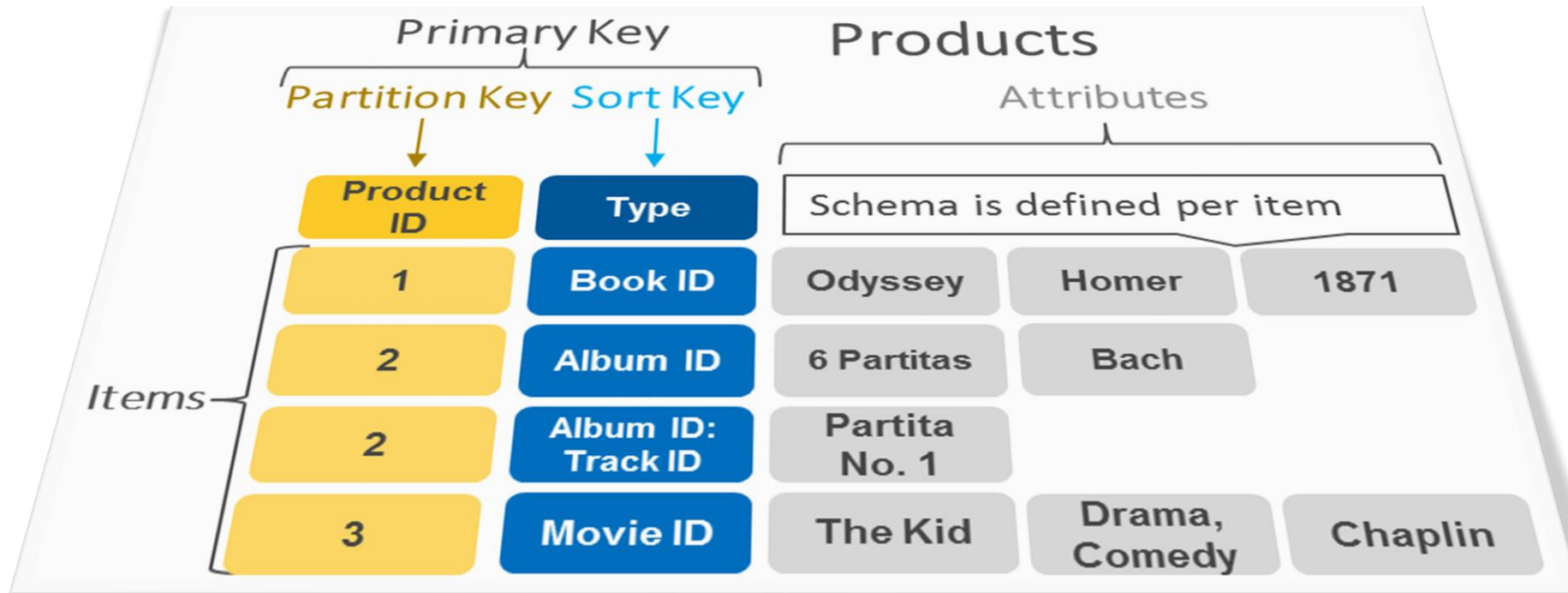
БД на основе пар «ключ-значение».

База данных на основе пар «ключ-значение» – это тип нереляционных баз данных, в котором для хранения данных используется простой метод «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.

Фрагмент массива ключ-значение.



Примеры использования БД на основе пар ключ-значение.

Хранилище сессий. Основанное на сессиях приложение (например, интернет-приложение) запускает сессию, когда пользователь входит в систему. Оно активно до тех пор, пока пользователь не выйдет из системы или не истечет время сессии. Данные сессии могут включать информацию профиля пользователя, сообщения, индивидуальные данные и темы, рекомендации, таргетированные рекламные кампании и скидки. Данные сессий всегда запрашиваются только по первичному ключу.

Корзина интернет-магазина. Во время праздничного сезона покупок сайт интернет-магазина может получать миллиарды заказов за считанные секунды. Используя базы данных на основе пар «ключ-значение», можно обеспечить необходимое масштабирование при существенном увеличении объемов данных и чрезвычайно интенсивных изменениях состояния.

Документная (документальная) БД.

Документная база данных – это тип нереляционных баз данных, предназначенный для хранения и запроса данных в виде документов в формате, подобном JSON. Документные базы данных позволяют разработчикам хранить и запрашивать данные в БД с помощью той же документной модели, которую они используют в коде приложения.

Документная модель хорошо работает в таких примерах использования, как каталоги, пользовательские профили и системы управления контентом, где каждый документ уникален и изменяется со временем.

Документные базы данных обеспечивают гибкость индексации, производительность выполнения стандартных запросов и аналитику наборов документов.

Пример документа документной БД.

```
[
  {
    "year" : 2013,
    "title" : "Turn It Down, Or Else!",
    "info" : {
      "directors" : [ "Alice Smith", "Bob Jones"],
      "release_date" : "2013-01-18T00:00:00Z",
      "rating" : 6.2,
      "genres" : ["Comedy", "Drama"],
      "image_url" : "http://ia.media-imdb.com/images/N/09ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@._V1_SX400_.jpg",
      "plot" : "A rock band plays their music at high volumes, annoying the neighbors.",
      "actors" : ["David Matthewman", "Jonathan G. Neff"]
    }
  },
  {
    "year": 2015,
    "title": "The Big New Movie",
    "info": {
      "plot": "Nothing happens at all.",
      "rating": 0
    }
  }
]
```

Примеры использования документных БД.

Управление контентом. Документная база данных – отличный выбор для приложений управления контентом, таких как платформы для блогов и размещения видео. При использовании документной базы данных каждая сущность, отслеживаемая приложением, может храниться как отдельный документ. Для внесения изменений в схему хранения данных нет необходимости обновлять ее и прерывать работу базы данных.

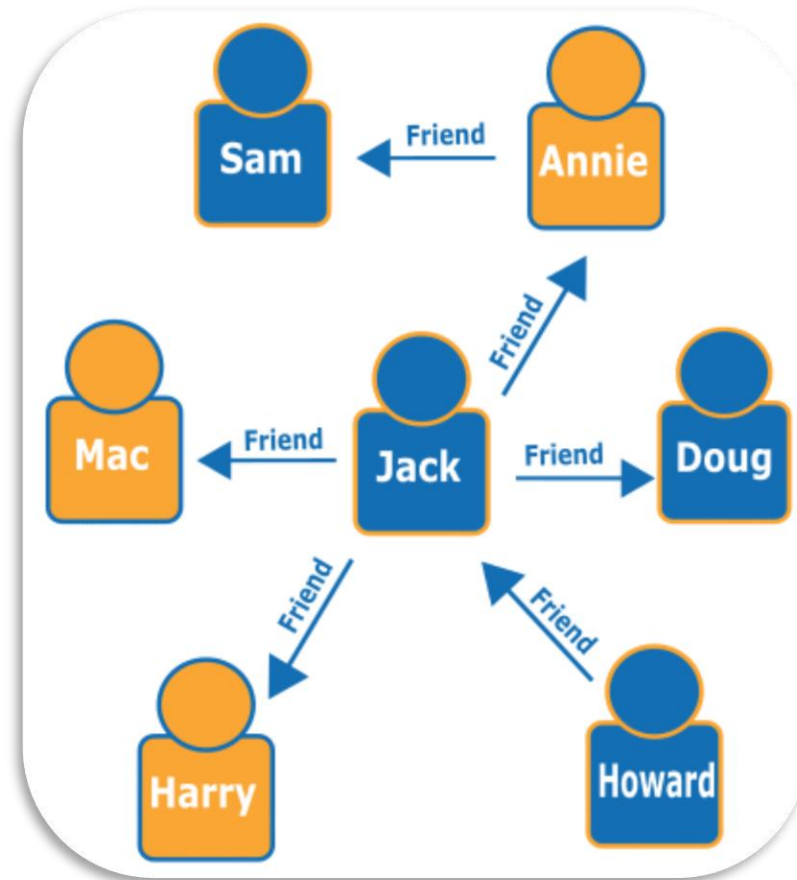
Каталоги. Документные базы данных эффективны для хранения каталожной информации. Например, в приложениях для интернет-коммерции разные товары обычно имеют различное количество атрибутов. При использовании документной базы данных атрибуты каждого товара можно описать в одном документе, что упрощает управление и повышает скорость чтения. Изменение атрибутов одного товара не повлияет на другие товары.

Графовые БД.

Графовые базы данных предназначены для хранения взаимосвязей и навигации в них. В графовых базах данных используются узлы для хранения сущностей данных и ребра для хранения взаимосвязей между сущностями. Ребро всегда имеет начальный узел, конечный узел, тип и направление. Ребра могут описывать взаимосвязи типа «родитель-потомок», действия, права владения и т. п.

Обход графа в графовой базе данных можно выполнять либо по определенным типам ребер, либо по всему графу. Обход соединений или взаимосвязей в графовых базах данных выполняется очень быстро, поскольку взаимосвязи между узлами не вычисляются во время выполнения запроса, а хранятся в базе данных.

Абстрактная схема графовой БД (граф социальной сети).



Примеры использования графовых БД.

Выявление мошенничества. Графовые базы данных позволяют выявлять сложные схемы мошенничества. Анализ взаимосвязей в графовых базах данных дает возможность обрабатывать финансовые операции и операции, связанные с покупками, практически в режиме реального времени. С помощью быстрых запросов к графу можно, например, определить, что потенциальный покупатель использует тот же адрес электронной почты и кредитную карту, которые уже использовались в известном случае мошенничества. Графовые базы данных также позволяют без труда обнаруживать определенные шаблоны взаимосвязей, например когда несколько человек связаны с одним персональным адресом электронной почты или когда несколько человек используют один IP-адрес, но проживают по разным физическим адресам.

Сервисы рекомендаций. Графовые базы данных – хороший выбор для рекомендательных приложений. Используя графовую базу данных, можно хранить в графе взаимосвязи между такими информационными категориями, как интересы покупателя, его друзья и история его покупок. Также, можно найти людей, у которых есть общий знакомый, но которые еще не знакомы друг с другом, и предложить им подружиться.

Чтение на дом

- Основы Data Science и Big Data. Стр. 18-40, глава Data Science в мире больших данных.
<https://drive.google.com/file/d/1RH8iydZ4oqCiNWhRivGJYhLIOKSqqisV/view?usp=sharing>
- Основы Data Science и Big Data. Стр. 183-199, глава Присоединяйтесь к движению noSQL.
- Конспект “Информационные системы и базы данных 2020”, Лекция 8. Или запись лекции: https://youtu.be/zrr_on-LndY

Спасибо за внимание!