Мультимодельные базы данных

Многомодельная база данных — это база данных, предназначенная для поддержки нескольких моделей данных в одной системе хранения данных. Это означает, что такая система может хранить, индексировать и запрашивать данные в нескольких моделях. Этот тип базы данных обеспечивает единый интерфейс для обеспечения согласованности, безопасности и доступа к данным, а также устраняет необходимость сложных преобразований и миграций между различными базами данных [1].

В настоящее время существует четыре основные модели баз данных: ключ-значение, семейство столбцов, документальные и графовые [2].

Базы данных ключ-значения имеют хранилище ключ-значение. Они позволяют разработчику приложения хранить данные без схемы. Эти данные обычно состоят из строки, которая представляет ключ, и фактических данных, которые считаются значением в отношениях «ключ-значение». Сами данные обычно представляют собой своего рода примитив языка программирования (строка, целое число, массив) или объект, который создаётся привязками языков программирования к хранилищу ключ-значение. Это заменяет необходимость в фиксированной модели данных и делает строгими требования к правильному форматированию данных без данных [3].

Графовая модель базы данных — это модель, в которой структуры данных для схемы и/или экземпляров моделируются как направленный, возможно, помеченный граф или обобщение структуры данных графа, где манипулирование данными выражается с помощью графо-ориентированных операций и конструкторов типов, а соответствующие ограничения целостности могут быть определены в структуре графа. [4]

База данных семейства столбцов - это база данных NoSQL, которая хранит данные с использованием столбцового подхода, в отличие от реляционных, которые упорядочивают данные по строкам. Данные, хранящиеся в базе данных семейства столбцов, выбираются вертикально, что делает частичное чтение более эффективным, поскольку загружается только набор атрибутов строки [5].

Вместо хранения данных в фиксированных строках и столбцах базы данных документов используют гибкие документы. Документ – это запись в базе данных документов. Документ обычно хранит информацию об одном объекте и любых связанных с ним метаданных. Документы хранят данные в парах поле-значение. Значения могут быть различных типов и структур, включая строки, числа, даты, массивы или объекты [6].

Далее представлены некоторые из наиболее популярных мультимодельных баз данных:

1. ArangoDB — это бесплатный менеджер баз данных с открытым исходным кодом, который поддерживает модели баз данных ключ-значение, документ и граф [7].

2. OrientDB поддерживает следующие модели баз данных: граф, документа, ключ-значение [8].

3. Azure Cosmos DB поддерживает следующие модели баз данных: документ, ключ-значение, граф [9].

Многомодельная обработка и оптимизация запросов. Несмотря на то, что многомодельные БД способны хранить данные в различных форматах (моделях), они не обеспечивают язык межмодельной обработки данных, межмодельную компиляцию или соответствующую оптимизацию многомодельных запросов. Напротив, многомодельная база данных пытается решить эту проблему путем разработки унифицированного языка запросов, который будет охватывать все поддерживаемые модели данных. Однако существующие языки запросов еще незрелы, и разработка полноценного языка запросов для многомодельных данных все еще остается открытой задачей. Тесно связанной проблемой является предложение подхода к определению оптимального плана запроса для эффективной оценки заданный межмодельный запрос.

Проектирование и оптимизация многомодельной схемы. Хороший дизайн схемы базы данных является важной частью, влияющей на многие аспекты, такие как эффективность обработки запросов, расширяемость приложения и т.д. В отличие от реляционных баз данных, базы данных NoSQL обычно используют значительно денормализованную физическую схему, которая требует дополнительного места. Следовательно, в многомодельных системах мы сталкиваемся с противоречивыми требованиями к различным моделям, и, таким образом, требуется новое решение для проектирования многомодельных схем, чтобы сбалансировать и найти компромисс между разнообразными требованиями к многомодельным данным. Даже вопрос существования схемы существенно различается: традиционные реляционные базы данных основаны на существовании заранее определенной схемы, тогда как базы данных NoSQL основаны на предположении об отсутствии схемы.

Мультимодельная эволюция. В целом, эффективное управление развитием схемы данных и распространением изменений в соответствующих частях системы базы данных, таких как экземпляры данных, запросы, индексы или даже стратегии хранения, является сложной задачей. В некоторых небольших приложениях компания может положиться на квалифицированный администратор базы данных для управления развитием данных и распространения изменений на другие затронутые части вручную. Но в большинстве случаев это сложная и подверженная ошибкам работа. В контексте многомодельных баз данных эта задача более тонкая и трудная.

Многомодельная расширяемость. Последней открытой проблемой является проблема расширяемости модели, которую можно рассматривать в нескольких аспектах. Во-первых, мы можем рассмотреть возможность расширения внутри модели, что означает расширение одной из моделей новыми конструкциями. Во-вторых, мы можем рассмотреть межмодельную расширяемость, которая добавляет новые конструкции, выражающие отношения между моделями. И в-третьих, мы можем обеспечить дополнительную расширяемость модели, которая включает добавление совершенно новой модели вместе с соответствующими данными и запросом [10].

**Категории взаимодействия**

Описание способов построения БД:

1. Polyglot persistence [11]
2. Лямбда архитектура [11] (?)
3. Мультимодельные СУБД на основе реляционной модели
4. Мультимодельные СУБД на основе документной модели
5. СУБД «без основной модели»

Методы общения

Краткое описание каждого

Проблемы и сложности

Текст

Литература

1. https://en.wikipedia.org/wiki/Multi-model\_database
2. The Multi-model Databases – A Review Ewa Płuciennik & Kamil Zgorzałek
3. Key-Value stores:a practical overview Marc Seeger Computer Science and MediaUltra-Large-Sites SS09 Stuttgart, Germany
4. Survey of graph database models Authors: Renzo Angles, Claudio Gutierrez
5. An Approach for Implementing Online Analytical Processing Systems under ColumnFamily Databases Abdelhak Khalil and Mustapha Belaissaoui
6. <https://www.mongodb.com/document-databases>
7. <https://docs.arangodb.com/3.11/concepts/data-models/>
8. <https://orientdb.org/docs/3.0.x/datamodeling/Tutorial-Document-and-graph-model.html>
9. <https://learn.microsoft.com/en-us/azure/cosmos-db/introduction>
10. Lu , J & Holubová , I 2019 , ' Multi-model Databases : A New Journey to Handle the Variety of Data ' , ACM Computing Surveys , vol. 52 , no. 3 , 55 . <https://doi.org/10.1145/3323214>
11. Polyglot database architectures = polyglot challenges