**ВСР - 2**

**Задание 2.2: Заполните таблицу "Преимущества и недостатки моделей данных"**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Модель данных** | **Преимущество** | **Недостатки** |
| 1 | *Иерархическая* | - Просто и легко понять.  - Эффективен для поиска и хранения данных.  - Хорошо подходит для иерархических данных, таких как организационные диаграммы, файловые системы и таксономии.  - Целостность данных поддерживается посредством принудительных отношений родитель-потомок. | - Ограниченная гибкость для обработки сложных отношений, таких как отношения «многие-ко-многим» (M:N).  - Потенциальная избыточность данных, поскольку дочерним узлам может потребоваться хранить повторяющуюся информацию о своем родительском узле.  - Не оптимизирован для доступа к данным посредством навигации между детьми и родителями или поиска неиерархических данных.  - Обновление или удаление данных может быть затруднено из-за жесткой иерархической структуры. |
| 2 | *Сетевая* | - Гибкость в представлении сложных отношений.  - Устраняет проблемы избыточности данных, обнаруженные в иерархической модели.  - Улучшена целостность данных за счет представления нескольких связей.  - Эффективен для поиска данных при прохождении связей. | - Повышенная сложность по сравнению с иерархической моделью.  - На производительность может повлиять сложность взаимоотношений.  - Обновление, удаление или вставка данных может оказаться более сложной задачей из-за взаимосвязанной структуры.  - Требует высокого уровня знаний для проектирования и обслуживания. |
| 3 | *Реляционная* | - Простое и интуитивно понятное представление данных.  - Очень гибкий для представления различных типов отношений.  - Обеспечивает надежную целостность данных посредством ограничений первичного и внешнего ключа.  - Простое манипулирование данными и их получение с помощью SQL.  - Широко поддерживается различными системами управления базами данных (СУБД). | - Может привести к проблемам с производительностью при работе с большими объемами данных или сложными запросами.  - Не оптимизирован для обработки иерархических или сетевых структур данных.  - Требует тщательного проектирования структур и связей таблиц, чтобы избежать избыточности данных и сохранить целостность данных. |
| 4 | *Объектно-ориентированная* | **- Естественное представление реальных объектов и их взаимосвязей**  **- Повторное использование кода (наследование)**  **- Инкапсуляция (сокрытие данных)**  **- Полиморфизм**  **- Модульность**  **- Поддержка сложных типов данных** | **- Сложность отображения на реляционные СУБД (объектно-реляционное несоответствие)**  **- Проблемы производительности**  **- Меньшая распространенность и поддержка по сравнению с реляционными СУБД**  **- Сложность запросов**  **- Более высокая сложность разработки (по сравнению с реляционной моделью) из-за необходимости тщательного проектирования классов и их взаимосвязей. - Возможные проблемы с масштабируемостью - Сложность отладки** |
| 5 | *NoSQL* | *Базы данных «ключ-значение»*  - Простота: Очень простая модель данных, легкая в понимании и использовании.  - Высокая скорость: Быстрые операции чтения и записи, так как поиск осуществляется по ключу.  - Масштабируемость: Легко масштабируются горизонтально, добавляя новые серверы.  - Гибкость: Поддерживают разные типы данных в качестве значений.  - Подходят для кэширования: Идеальны для кэширования данных, так как обеспечивают быстрый доступ.  *Графовые базы данных*  - Естественное представление связей: Идеально подходят для моделирования данных, где связи имеют первостепенное значение (социальные сети, сети знаний, рекомендации).  - Высокая производительность при запросах к связям: Быстрые запросы для поиска связанных данных, обхода графа.  - Гибкость схемы: Легко добавлять новые типы узлов и связей без изменения существующей структуры.  - Анализ связей: Подходят для анализа сложных связей и зависимостей.  *Колоночные базы данных*  - Высокая производительность аналитических запросов: Оптимизированы для выполнения аналитических запросов (OLAP), таких как агрегация и фильтрация данных.  - Эффективное сжатие данных: Хранение данных по столбцам позволяет эффективно сжимать однородные данные.  - Обработка больших объемов данных: Хорошо подходят для хранения и обработки больших объемов данных (Data Warehouses).  - Снижение I/O операций: Чтение только необходимых столбцов уменьшает количество операций ввода-вывода. | *Базы данных «ключ-значение»*  - Ограниченные возможности запросов: Не поддерживают сложные запросы, так как нет структуры, как в реляционных базах данных.  - Отсутствие ACID-транзакций: Обычно обеспечивают только согласованность в конечном итоге (eventual consistency).  - Сложность связывания данных: Сложно моделировать сложные взаимосвязи между данными.  - Ограниченные возможности по анализу данных: Анализ данных может быть затруднен из-за отсутствия структуры.  *Графовые базы данных*  - Ограниченная сфера применения: Не подходят для задач, где связи не играют ключевой роли.  - Сложность запросов: Требуют знания специальных языков запросов (например, Cypher).  - Трудности с масштабированием некоторых типов запросов: Некоторые типы графовых запросов могут быть сложными в масштабировании.  - Меньшая зрелость технологии: Менее распространены и изучены по сравнению с реляционными и некоторыми другими типами NoSQL баз данных.  *Колоночные базы данных*  - Низкая производительность транзакционных операций: Не оптимизированы для выполнения частых операций записи и обновления данных (OLTP).  - Сложность изменения схемы: Изменение схемы может быть сложным и дорогостоящим.  - Не подходят для хранения слабоструктурированных данных: Эффективны только для хорошо структурированных данных.  - Сложность реализации некоторых типов запросов: Некоторые типы запросов, требующие объединения данных из разных столбцов, могут быть сложными в реализации.  - Требуют специальных навыков для администрирования. |
| 6 | *NewSQL* | - Масштабируемость (горизонтальная): Способны масштабироваться горизонтально, распределяя нагрузку на несколько узлов.  - ACID-транзакции: Поддерживают ACID-транзакции для обеспечения надежности и целостности данных.  - SQL-совместимость: Используют SQL (или его подмножество), что упрощает разработку и миграцию для разработчиков, знакомых с реляционными СУБД.  - Производительность: Часто показывают высокую производительность, сопоставимую с NoSQL, при одновременной поддержке транзакций.  - Консистентность: Обеспечивают консистентность данных.  - Устраняют ограничения традиционных реляционных СУБД в плане масштабируемости. | - Сложность реализации: Разработка NewSQL баз данных — сложная задача, и их реализация может быть более сложной, чем у традиционных реляционных СУБД или NoSQL.  - Небольшое количество реализаций (на данный момент): На рынке представлено меньше реализаций NewSQL по сравнению с другими типами баз данных.  - Специфические требования к оборудованию и настройке  - Обучение и адаптация: Разработчикам и администраторам может потребоваться дополнительное обучение и адаптация, чтобы эффективно работать с NewSQL СУБД.  - Могут быть не оптимальны для очень специфических задач |