Транзакция – это последовательность действий с базой данных, которые либо все выполняются успешно, либо не выполняется ни одно из них.

Свойства транзакции (требования, ACID)

1. Атомарность – ни одна транзакция не выполнится частично.

2. Согласованность – до и после выполнения транзакции база данных находится в согласованном состоянии.

3. Изоляция – параллельно выполняемые транзакции не должны мешать друг другу.

4. Долговечность (надежность) – если транзакция завершена, то ее результат сохранится, даже если сразу после этого произошел сбой.

Проблемы конкурирующих транзакций:

1. Проблема потерянного обновления – когда несколько транзакций одновременно прочитали меняют одни и те же данные, и сохраняется только результат последнего обновления.

2. Проблема грязного чтения – одна транзакция прочитала данные, измененные другой транзакцией, которая после выполнения будет отменена.

3. Проблема неповторяемого чтения – одна транзакция прочитала данные, потом их изменила другая транзакция, но первая была выполнена на основании разных данных.

4. Проблема фантомного чтения – одна транзакция просматривает в таблице все данные, во время ее выполнения другая транзакция добавляет или удаляет данные. Следовательно первая транзакция выполнена для разных данных

Блокировки

Явные блокировки – блокировки, прописанные в тексте команды

Неявные блокировки – блокировки, которые накладываются самой СУБД

Объект блокировки – строка, таблица, БД

Монопольные – блокируют все виды доступа, коллективные – блокируют только изменение

Уровни изоляции транзакций

1. Незавершенное чтение – изменять данные может только одна транзакция. Решает проблему потерянного обновления. Чтение неподтвержденных данных

2. Завершенное чтение – если одна транзакция меняет данные, то другая транзакция не может их прочитать. Решает проблему грязного чтения

3. Повторяемое чтение – если одна транзакция читает данные, то другая транзакция не может их изменить. Решает проблему неповторяемого чтения

4. Сериализуемость – если одна транзакция обрщается к данным, то другая транзакция не может добавить или удалить данные, которые могут быть прочитаны этой транзакцией. Избавляет от фантомного чтения.

Безопасная система – это система, которая посредством особых механизмов защиты контролирует доступ к данным так, чтобы только обладающие определенными полномочиями лица смогли получить доступ к чтению, изменению, добавлению или удалению данных.

CRUD – create, read, update, delete

Классы безопасности

1. D – класс, не относящийся ни к одному из классов A, B, C

2. Класс C – 1) имеет систему идентификации и аутентификации

Идентификация – это фиксация в системе идентификатора пользователя.

Аутентификация – это процесс проверки введенного идентификатора с тем, что есть в системе

Авторизация – это предоставление прав доступа пользователю.

2) авторизация - избирательный (дискреционный) контроль доступа. Осуществляется с помощью таблицы, где строки – это субъекты, а столбцы – это объекты. Тогда для каждого субъекта и для каждого объекта устанавливается уровень доступа.

Класс C1 – субъект может быть группой пользователей, C2 – субъект может быть только одним пользователем

3. Класс B – 1) имеет систему идентификации и аутентификации

2) авторизация – мандатный контроль доступа. Каждому объекту и субъекту сопоставляется метка секретности. Субъект может читать объекты своего уровня секретности и ниже, а писать только в свой уровень секретности и выше.

B2 — снабжаем метками секретности не только объекты данных, но и все ресурсы, которые прямо или косвенно доступны субъекту.

B3 — добавляется журналирование всех действий по получению доступа

4. Класс А - ведется учет всех действий разработчиков, документирование. и проверяется нет ли уязвимостей во время процесса разработки.

Распределенная база данных – это набор логически связанных между собой разделяемых данных и их описаний, которые физически распределены по разным вычислительным узлам.

Стратегии размещения данных в системе:

1. Раздельное (фрагментированное) размещение – каждый фрагмент существует ровно в одном экземпляре и лежит на отдельном узле.

Достоинства: не тратим много памяти

Недостатки: надежность (если 1 узел потерял данные, то их больше нигде нет)

2. Размещение с полной репликацией – каждый узел хранит реплику БД

Достоинства: надежность

Недостатки: память, синхронизация (приходится передавать большие объемы данных при обвнолении)

3. Размещение с выборочной репликацией. Для каждого узла определяем нужное количество реплик

Прозрачность – это скрытие от конечного пользователя деталей реализации. Прозрачность фрагментации, расположения, репликации, контроля доступа.

12 правил Дейта для распределенных баз данных

1) Локальная автономность. Локальные данные принадлежат локальным владельцам и сопровождаются локально. Надежность, безопасность и производительность обеспечиваются программным и аппаратным обеспечением этого узла.

2) Отсутствие опоры на центральный узел. Нет ни одного узла, без которого система не может функционировать

3) Непрерывное функционирование. Никогда не должно возникать необходимости планово остановить систему. Например, нельзя каждые два часа останавливать систему и ее синхронизировать

4) Независимость от расположения (прозрачность расположения) – доступ должен получать доступ к данным независимо от того, где они расположены

5) Независимость фрагментации. Пользователь должен получать доступ ко всем данным независимо от того, как они фрагментированы

6) Независимость от репликации. Пользователь не должен нуждаться о каких-либо сведениях о репликация. За актуальность всех копий отвечает система.

7) Обработка распределенных запросов. Система должна поддерживать обработку запросов к данным, оказавшихся на разных узлах.

8) Обработка распределенных транзакций

9) Независимость от типа оборудования

10) Независимость от сетевой архитектуры

11) Независимость от операционной системы

12) Независимость от типа СУБД (должна быть гетерогенной, реляционная и

объектно-ориентированная модели должны быть соединимы в одну бд без

проблем)

Предпосылки к появлению NoSQL:

1) Увеличение объемов данных

2) Сильная взаимосвязанность данных

3) Слабая структурированность данных

Что отличает базу данных от базы знаний?

У базы знаний, в отличие от базы данных, есть механизм вывода новых знаний.