**Транзакции и блокировки в базах данных**

1. Транзакция

**Определение:** Транзакция - это логическая единица работы с базой данных, которая переводит БД из одного согласованного состояния в другое.

**Характеристики:**

* Атомарность (выполняется полностью или не выполняется вообще)
* Последовательность операций
* Изменение состояния БД

**Пример:** Перевод денег между счетами:

sql

Copy

Download

BEGIN TRANSACTION;

UPDATE accounts SET balance = balance - 100 WHERE id = 1;

UPDATE accounts SET balance = balance + 100 WHERE id = 2;

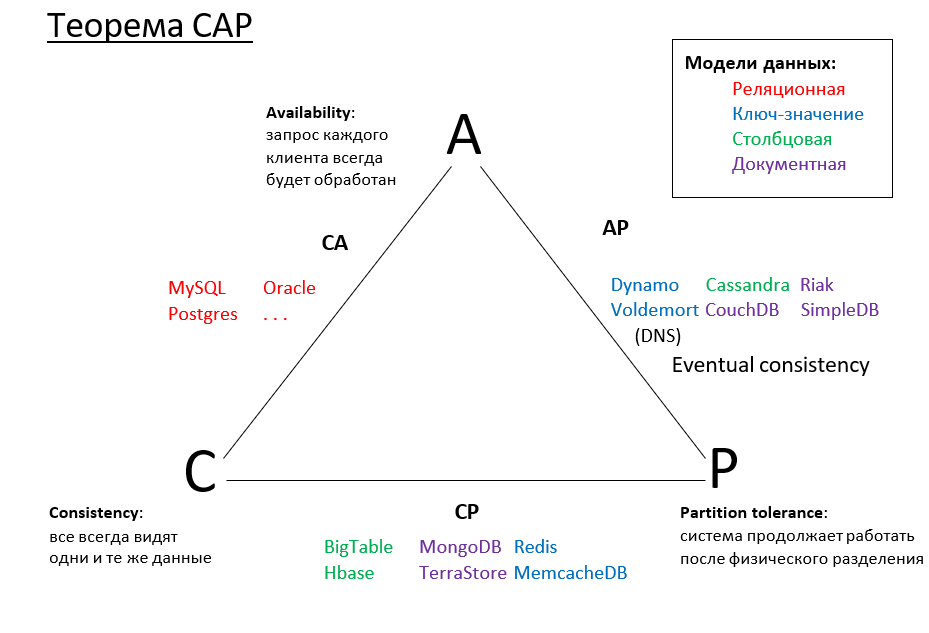
COMMIT;

2. Виды транзакций

1. **Явные транзакции** - определяются программистом с помощью BEGIN/COMMIT
2. **Неявные транзакции** - каждая команда выполняется как отдельная транзакция
3. **Распределённые транзакции** - затрагивают несколько БД на разных серверах
4. **Вложенные транзакции** - транзакции внутри других транзакций

3. ACID свойства транзакций

1. **Atomicity (Атомарность)** - транзакция выполняется как единое целое
2. **Consistency (Согласованность)** - БД переходит из одного согласованного состояния в другое
3. **Isolation (Изолированность)** - параллельные транзакции не влияют друг на друга
4. **Durability (Долговечность)** - результаты успешной транзакции сохраняются после сбоев

4. Теорема CAP  


**CAP-теорема** утверждает, что распределённая система может гарантировать только два из трёх свойств:

* **Consistency** (согласованность)
* **Availability** (доступность)
* **Partition tolerance** (устойчивость к разделению)

**Типы систем:**

* CA (традиционные SQL БД)
* CP (MongoDB, HBase)
* AP (Cassandra, DynamoDB)

5. Журнал транзакций (Transaction Log)

**Назначение:**

* Восстановление после сбоев
* Поддержка отката транзакций
* Репликация данных

**Принцип работы:** Все изменения сначала записываются в журнал (WAL - Write Ahead Log), затем в саму БД.

6. Блокировки

**Типы блокировок:**

1. **Разделяемые (Shared)** - для чтения
2. **Эксклюзивные (Exclusive)** - для записи
3. **Намеренные (Intent)** - указывают намерение установить блокировку

**Гранулярность блокировок:**

* Таблица
* Страница
* Строка
* Поле

7. Проблема параллелизма

**Основные проблемы:**

1. **Потерянное обновление** (Lost Update) **Суть проблемы:**  
   Два параллельных транзакции перезаписывают изменения друг друга без учёта предыдущих модификаций
2. **"Грязное" чтение** (Dirty Read) **Суть проблемы:**  
   Чтение незафиксированных данных, которые могут быть отменены другой транзакцией
3. **Неповторяющееся чтение** (Non-repeatable Read) **Суть проблемы:**  
   Повторное чтение тех же данных в рамках одной транзакции возвращает разные значения из-за изменений другой транзакции.
4. **Фантомное чтение** (Phantom Read) **Суть проблемы:**  
   Появление новых строк (фантомов) между повторными чтениями в одной транзакции.
5. Сравнительная таблица проблем

| **Проблема** | **Уровень изоляции, где возникает** | **Решающий уровень изоляции** |
| --- | --- | --- |
| Потерянное обновление | READ UNCOMMITTED | REPEATABLE READ |
| Грязное чтение | READ UNCOMMITTED | READ COMMITTED |
| Неповторяющееся чтение | READ COMMITTED | REPEATABLE READ |
| Фантомное чтение | REPEATABLE READ | SERIALIZABLE |

**Важно:** В разных СУБД (PostgreSQL, MySQL, Oracle) реализация уровней изоляции может отличаться!

8. Deadlock (Взаимная блокировка)

**Определение:** Ситуация, когда две или более транзакции ожидают освобождения ресурсов, заблокированных друг другом.

**Пример:**

Copy

Download

Транзакция 1: блокирует A, запрашивает B

Транзакция 2: блокирует B, запрашивает A

**Способы обработки:**

* Обнаружение и отмена одной из транзакций
* Предотвращение (упорядочивание блокировок)
* Ожидание с таймаутом

9. Проблемы одновременного доступа к данным

1. **Read Skew** - непоследовательное чтение нескольких объектов
2. **Write Skew** - две транзакции читают одни данные, но пишут разные
3. **Phantom Reads** - появление новых строк при повторном чтении

10. Уровни изоляции транзакций

1. **Read Uncommitted** - чтение незафиксированных данных
2. **Read Committed** - чтение только зафиксированных данных
3. **Repeatable Read** - гарантирует повторяемость чтений
4. **Serializable** - полная изоляция, как последовательное выполнение

11. Установка уровня изоляции транзакций

**SQL стандарт:**

sql

Copy

Download

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

**Примеры для СУБД:**

* **MySQL:**

sql

Copy

Download

SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;

* **SQL Server:**

sql

Copy

Download

SET TRANSACTION ISOLATION LEVEL SNAPSHOT;

**На уровне соединения (JDBC):**

java

Copy

Download

connection.setTransactionIsolation(Connection.TRANSACTION\_REPEATABLE\_READ);