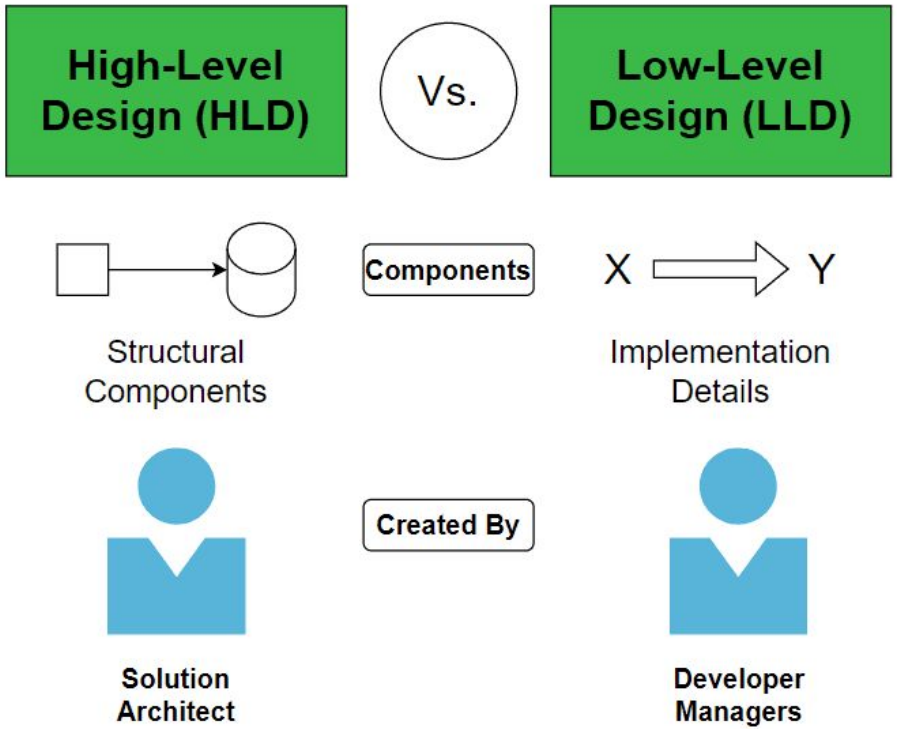
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Системный дизайн современных приложений

Лекция №5
Компоненты HLD: базы данных

HLD vs LLD


High-Level Design (HLD)
Vs.
Low-Level Design (LLD)
Components
X
Y
Structural Components
Implementation Details
Created By
Solution Architect
Developer Managers

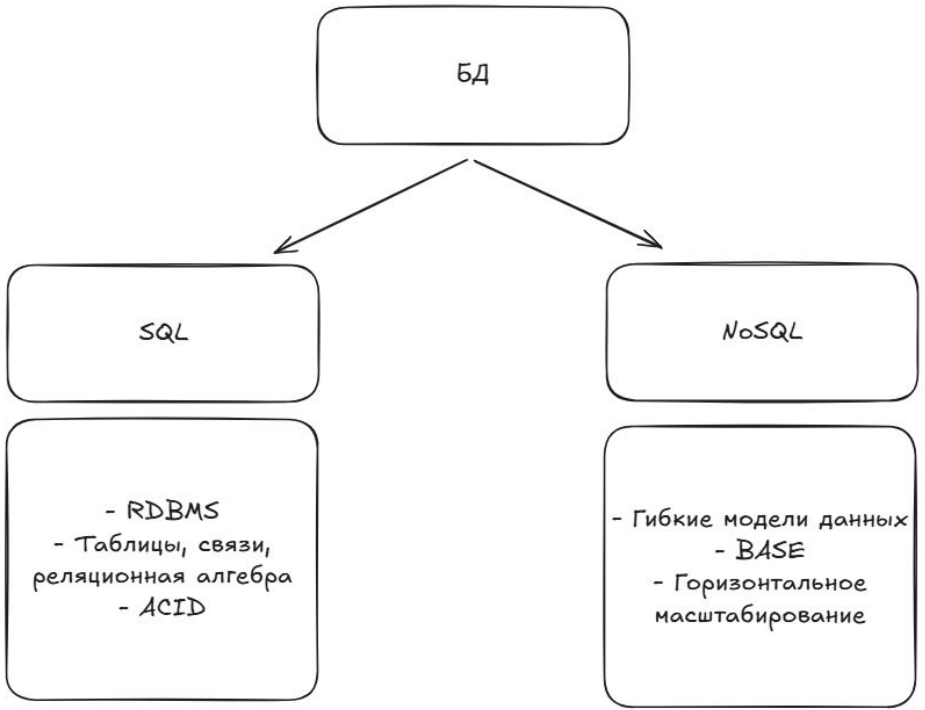
Основные аспекты SD

1. Масштабируемость
2. Производительность
3. Надежность
   a. Отказоустойчивость
   b. Доступность
4. Безопасность
5. Адаптивность
6. Управляемость и мониторинг
7. Интеграции

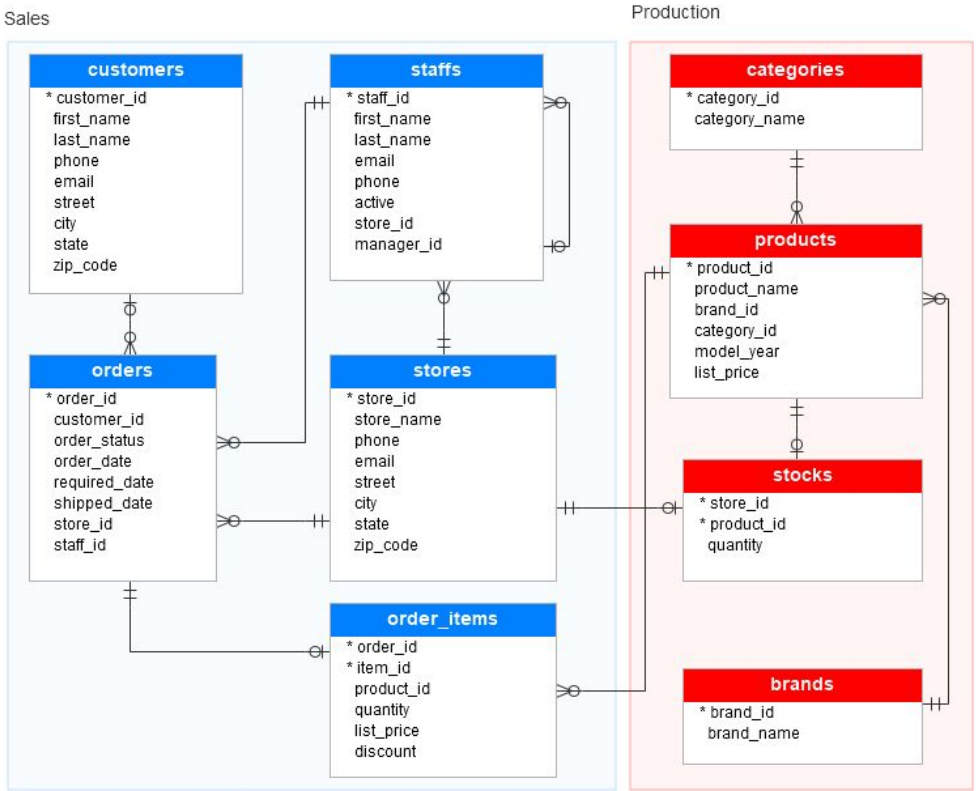
Основные сущности

1. *Клиент*
2. *Сервис*
3. *Интеграция*
4. **База данных**

Базы данных


БД
SQL
NoSQL
- RDBMS
- Таблицы, связи, реляционная алгебра
- ACID
- Гибкие модели данных
- BASE
- Горизонтальное масштабирование

SQL


Sales
Production
customers
* customer_id
first_name
last_name
phone
email
street
city
state
zip_code
staffs
* staff_id
first_name
last_name
email
phone
active
store_id
manager_id
categories
* category_id
category_name
products
* product_id
product_name
brand_id
category_id
model_year
list_price
orders
* order_id
customer_id
order_status
order_date
required_date
shipped_date
store_id
staff_id
stores
* store_id
store_name
phone
email
street
city
state
zip_code
stocks
* store_id
* product_id
quantity
order_items
* order_id
* item_id
product_id
quantity
list_price
discount
brands
* brand_id
brand_name

Транзакция

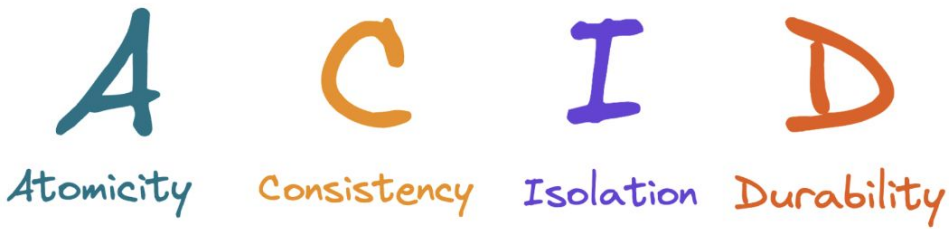
Транзакция — упорядоченное множество операций, переводящих базу данных из одного согласованного состояния в другое
BEGIN TRANSACTION … COMMIT

Зачем?
Решение проблем
1. при сбоях БД
2. при конкурентном доступе до данных в БД

Когда НЕ нужны?
1. Читаем/обновляем только одну запись за операцию
2. Нет конкурентного доступа до данных

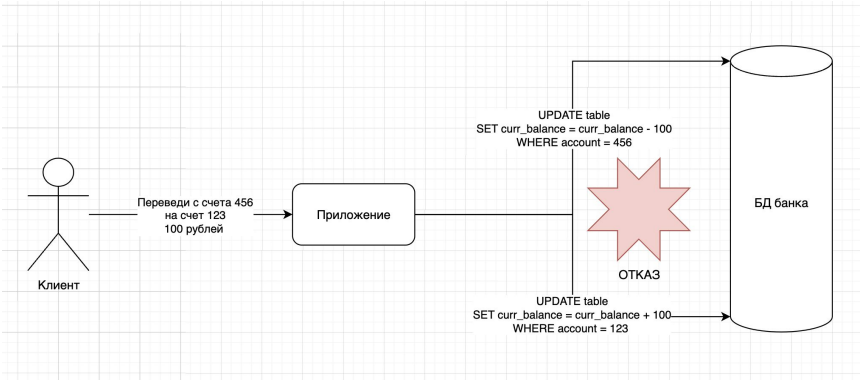
ACID

Набор гарантий безопасности, которые обеспечивают транзакции

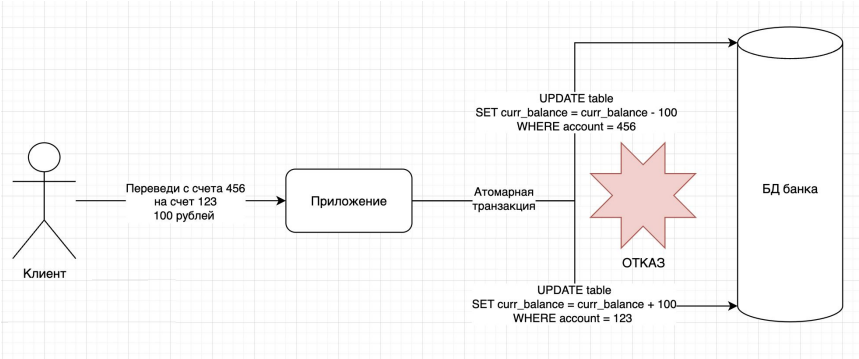


Atomicity (атомарность)

Транзакция будет выполнена либо полностью (commit),
либо не будет выполнена совсем (abort + rollback)



Итог: 100 рублей растворились



Итог: вернули клиенту ошибку

Consistency (согласованность)

1. Атомарность, изоляция и сохраняемость — свойства базы данных, в то время как согласованность — свойство приложения.
2. По словам Клеппмана добавили в мнемонику для красоты.

Про правильность данных:

Целостность данных (например, внешние ключи, уникальность, проверки CHECK).

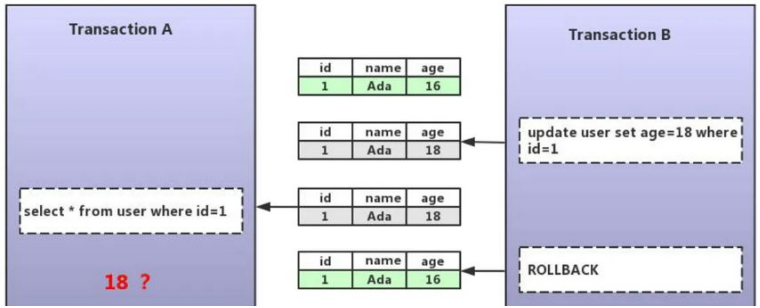
Бизнес-правила (например, «баланс счета не может быть отрицательным»).

Isolation (изоляция)

Конкурентно выполняемые транзакции не могут помешать друг другу

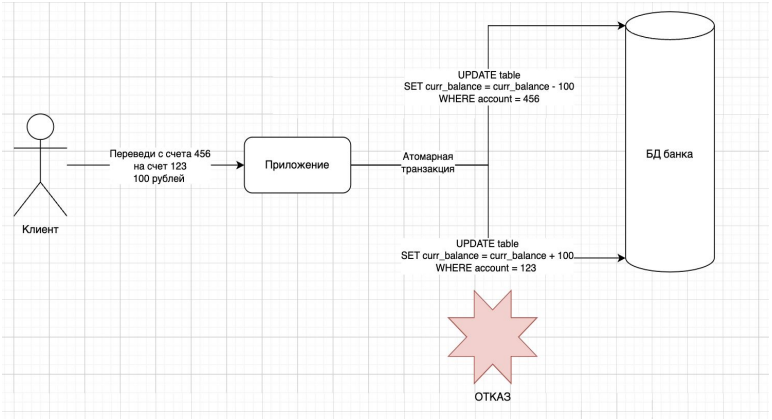
Примеры:

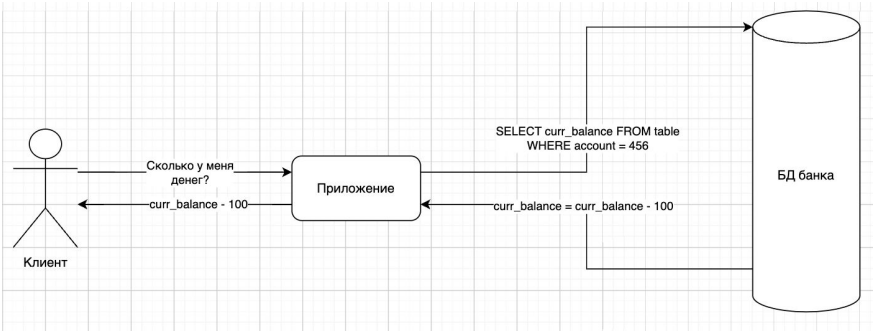
- Phantom Read: появились/исчезли строки между запросами
- Dirty Read: читаем незакомиченные данные
- Non-Repeatable Read: данные изменились между чтениями



Durability (Устойчивость)

Завершенные (зафиксированные) транзакции не потеряются





1) Закоммитили транзакцию
2) БД упала

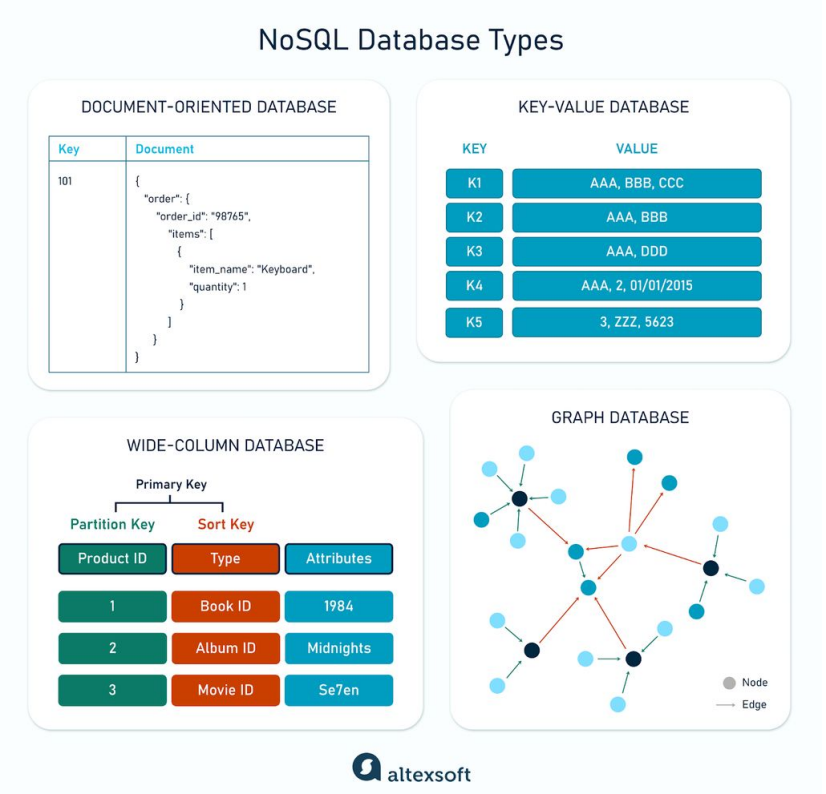
1) БД восстановили
2) Данные актуальные

NoSQL

NoSQL != No SQL
NoSQL == Not Only SQL

Зачем создали?

1. Горизонтальное масштабирование
2. Новые модели данных
3. Гибкость схемы данных
4. Новые модели консистентности



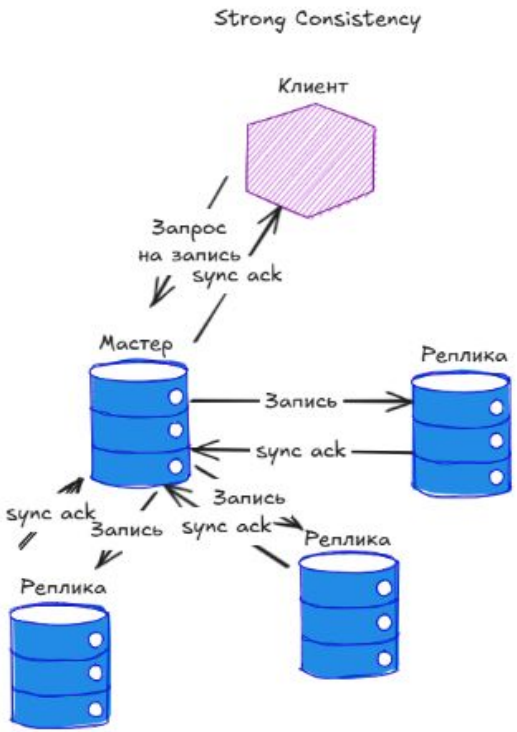
BASE

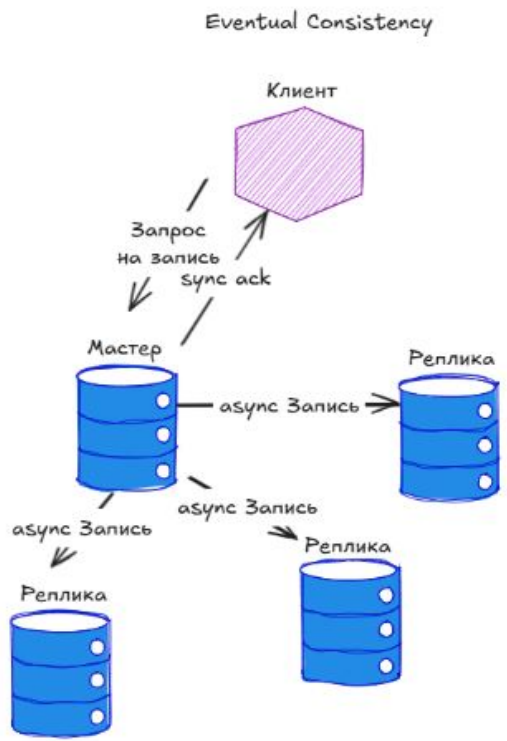
1 млн RPS
Транзакции блокируют запись
Что делать?

- Basically Available — система всегда отвечает, даже с устаревшими данными.
- Soft state — данные могут временно быть несогласованными.
- Eventually consistent — согласованность достигается со временем.

Виды согласованности

Характеристика	Strong	Eventual
Гарантия согласованности	Данные одинаковые на всех узлах	Разные узлы могут временно содержать разные версии данных
Задержка обновлений	Высокая – каждый запрос ждёт подтверждения от всех узлов	Низкая – изменения распространяются асинхронно
Пример использования	Банковские транзакции	Социальные сети, поисковые системы, кэширование
Инфраструктурные особенности	Требует синхронной репликации и блокировок	Использует асинхронные механизмы репликации



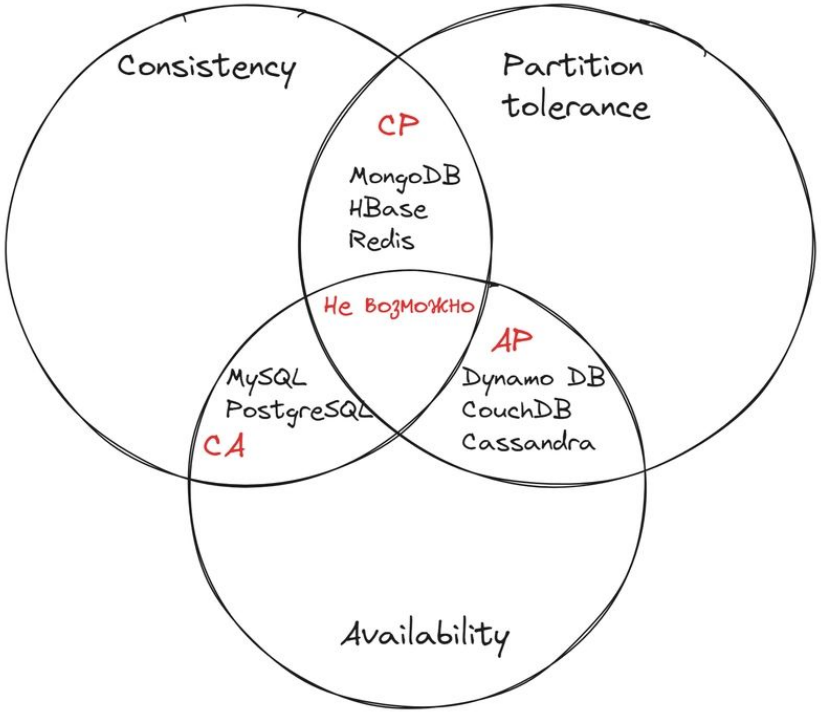


CAP-теорема

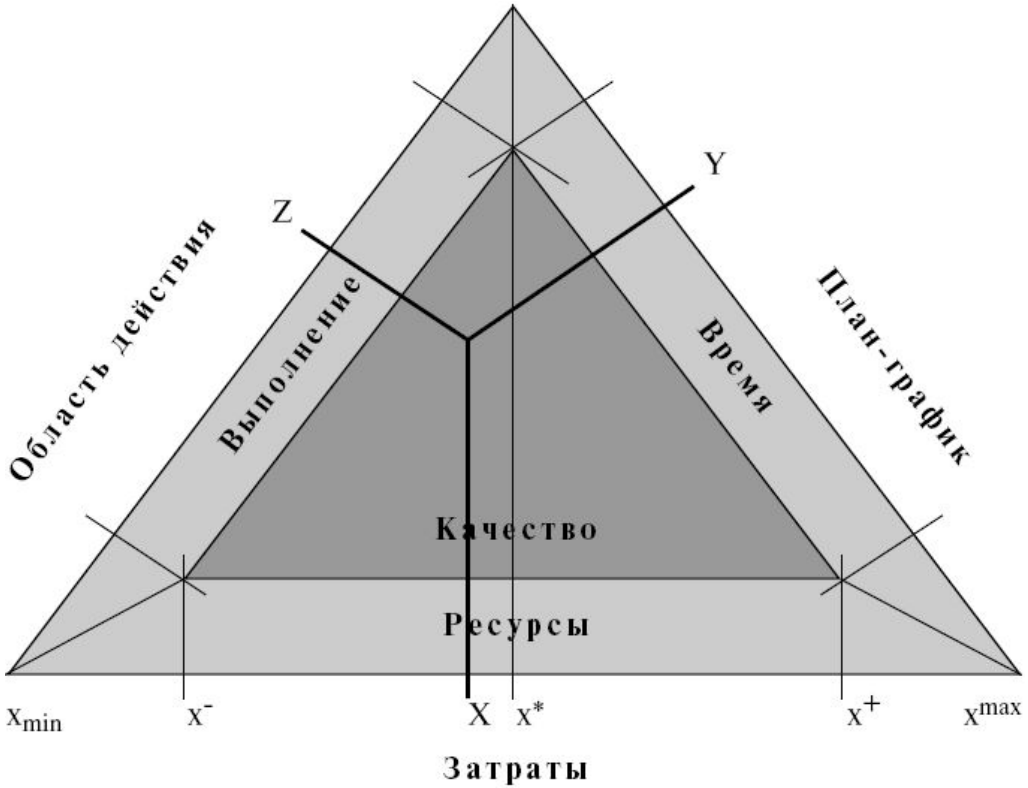
CAP-теорема про компромиссы

Распределенные системы могут обеспечивать только 2 свойства из 3-х:

1. Согласованность (Consistency) *(!= Согласованность в ACID (!))*
   a. **В любой момент времени на узлах одни и те же данные.**
2. Доступность (Availability).
   a. **Система всегда отвечает на запросы.**
3. Устойчивость к разделению (Partition tolerance).
   a. **Если между узлами нет связи, система работает.**

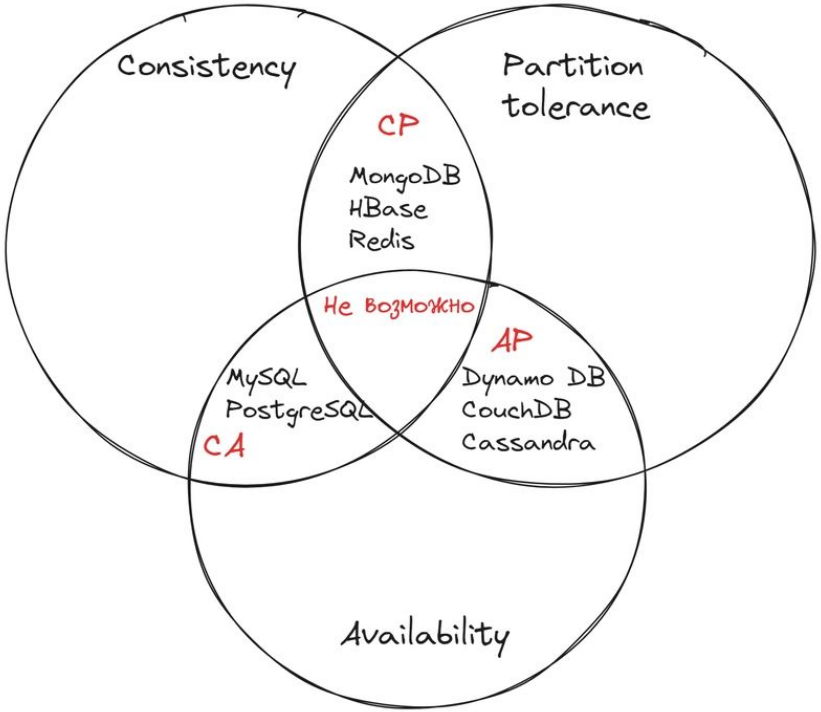


CAP-теорема аналогия



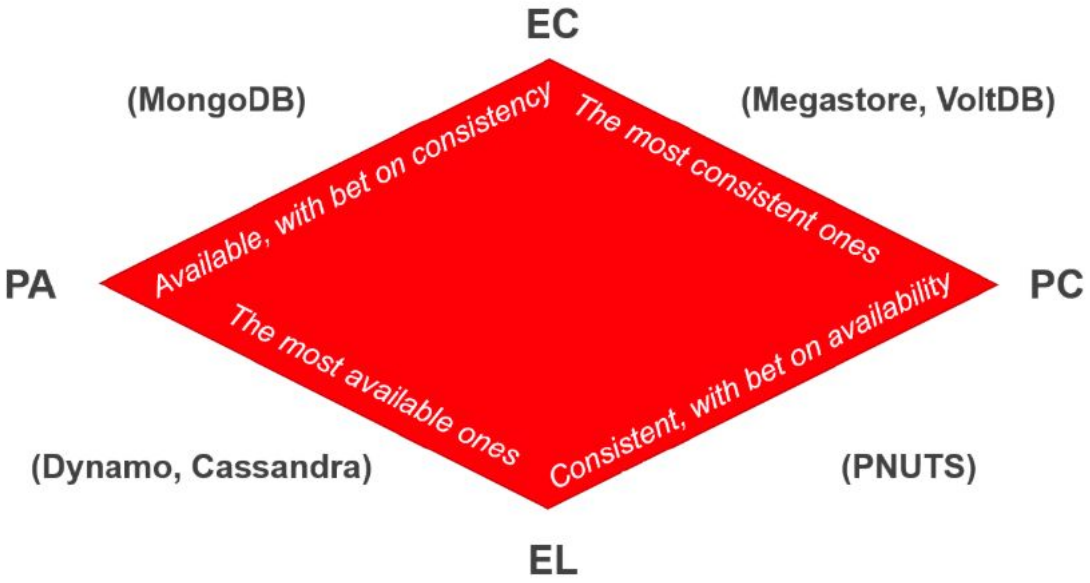
Проблемы CAP-теоремы

1. “Идеальный вольтметр”
2. В распределенных системах сетевые сбои неизбежны, а значит P - обязательно
3. Игнорирование задержек

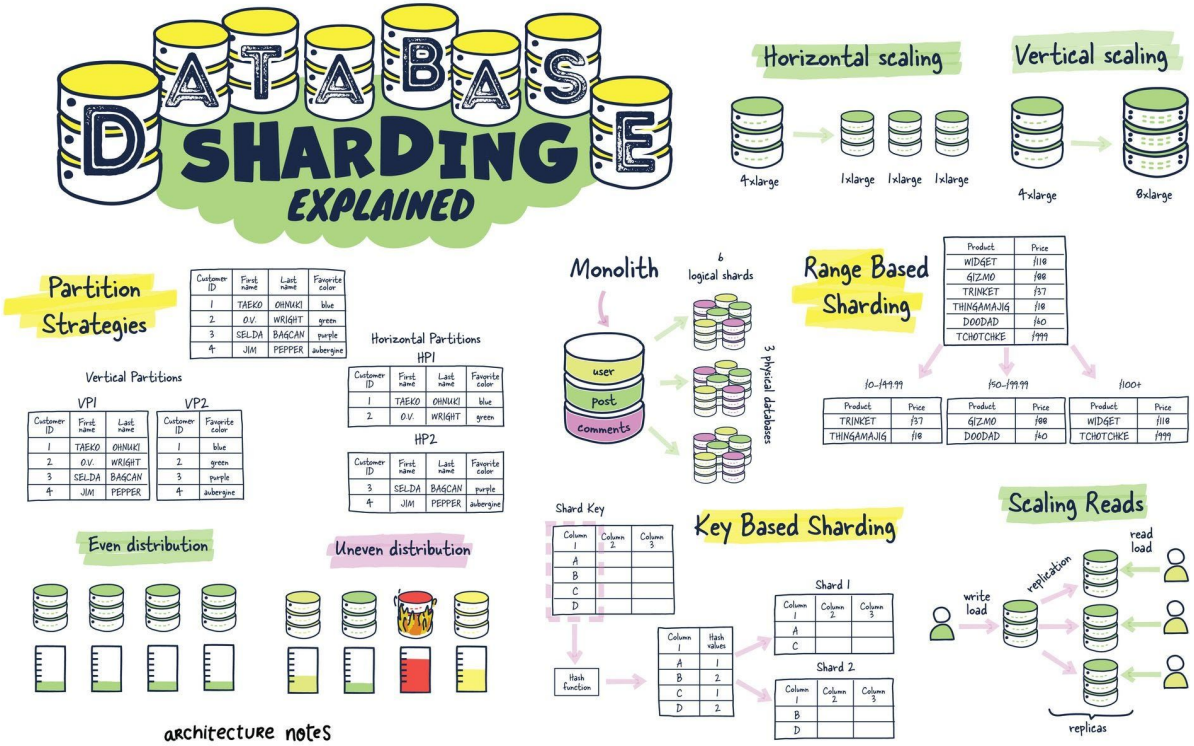


PACELC

if **P**artition tolerance -> (**A**vailability or **C**onsistency) **E**lse (**L**atency or **C**onsistency)



Репликация, шардирование, партиционирование



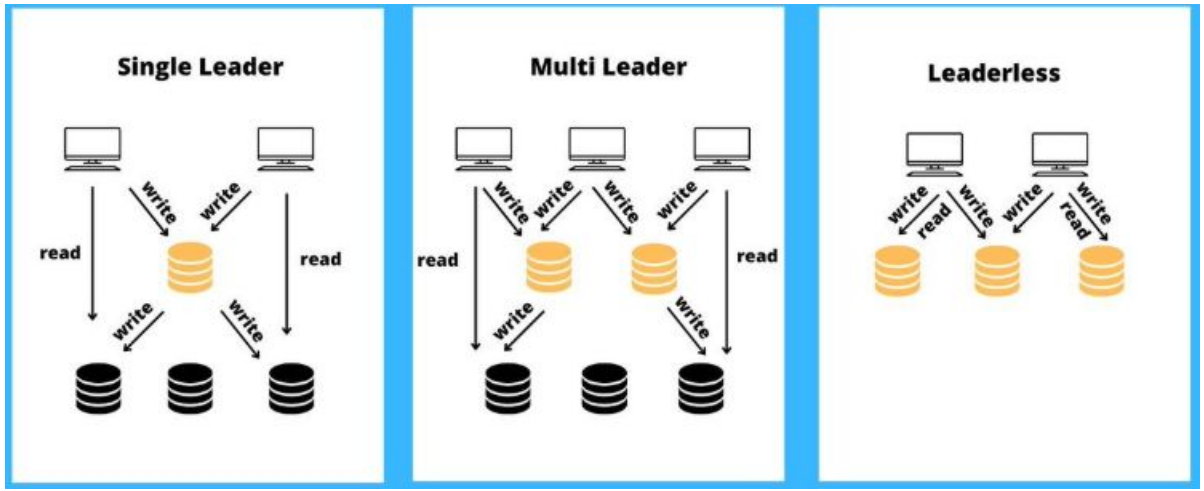
Репликация

Зачем

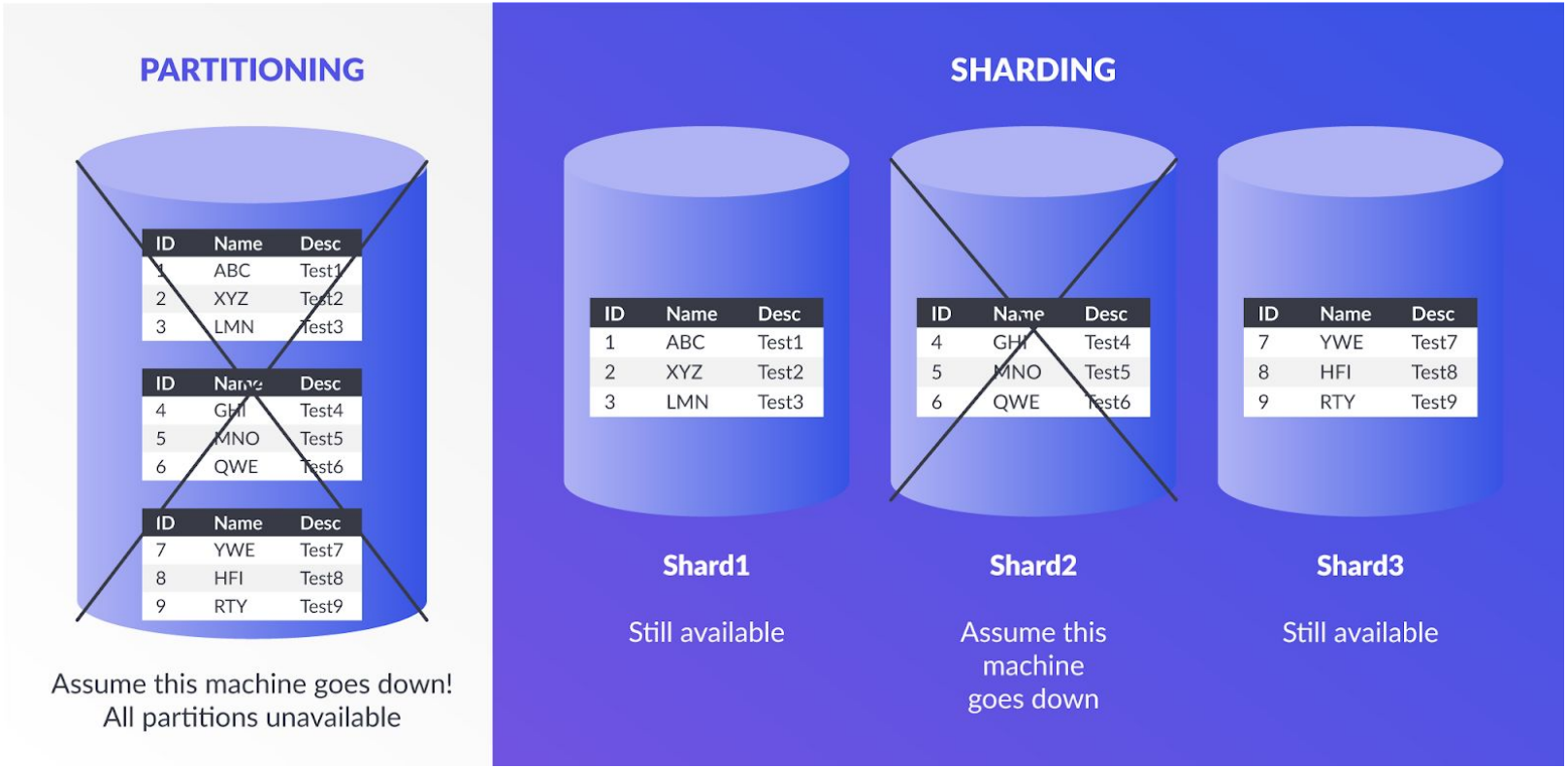
1. Повышение доступности
2. Улучшение производительности
3. Геораспределение

Виды

1. Single-Leader
2. Multi-Leader
3. Leaderless



Шардинг и партиционирование



Шардинг и партиционирование

Шардинг (Sharding)

Физическое разделение данных между разными серверами (например, по user_id).

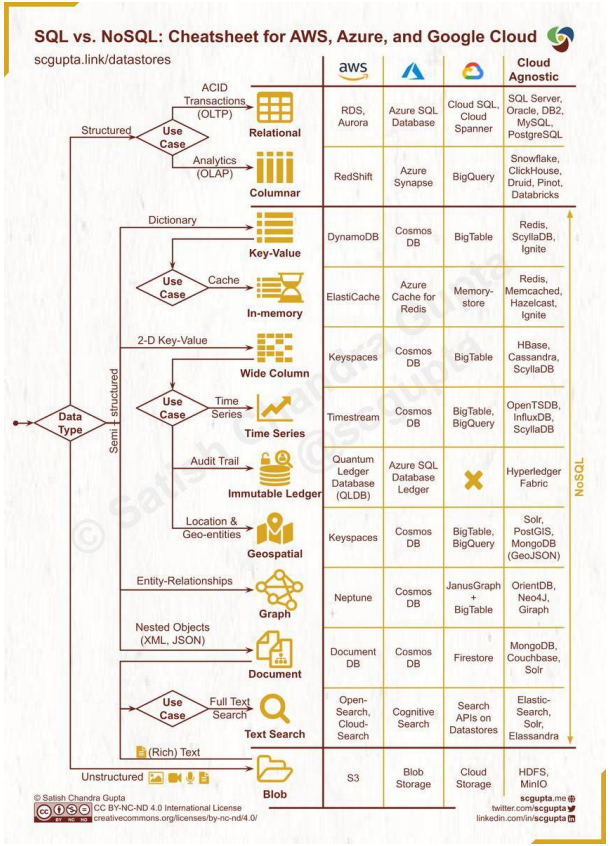
Цель: **Распределить** нагрузку на запись и хранить большие объемы данных.

Партиционирование (Partitioning)

Логическое разделение данных внутри одной БД (например, по датам).

Цель: **Ускорить** запросы и упростить управление данными.

Алгоритм выбора БД



Ряд вопросов:

1. Что за данные?
2. Какой формат данных?
3. Какие отношения между данными? (ERD)
4. Какие сценарии использования данных?
5. Типовой ли проект?
6. Какие функции доступны из коробки?
7. Насколько популярна?
8. Какая лицензия?
9. Легко сопровождать и настраивать?
10. Что если данные изменятся?
11. (*) Есть ли сертификация/регистрация в ЕРРП?

\+ https://docs.google.com/viewerng/viewer?url=https://datafinder.ru/files/Cheat-Sheet-SUBD.pdf

Тренировка №1

Кейс: фото-апп

Ряд вопросов:

1. Что за данные?
2. Какой формат данных?
3. Какие отношения между данными? (ERD)
4. Какие сценарии использования данных?
5. Типовой ли проект?
6. Какие функции доступны из коробки?
7. Насколько популярна?
8. Какая лицензия?
9. Легко сопровождать и настраивать?
10. Что если данные изменятся?
11. (*) Есть ли сертификация/регистрация в ЕРРП?

\+ https://docs.google.com/viewerng/viewer?url=https://datafinder.ru/files/Cheat-Sheet-SUBD.pdf

Тренировка №2

Кейс: авиасейлс

Ряд вопросов:

1. Что за данные?
2. Какой формат данных?
3. Какие отношения между данными? (ERD)
4. Какие сценарии использования данных?
5. Типовой ли проект?
6. Какие функции доступны из коробки?
7. Насколько популярна?
8. Какая лицензия?
9. Легко сопровождать и настраивать?
10. Что если данные изменятся?
11. (*) Есть ли сертификация/регистрация в ЕРРП?

\+ https://docs.google.com/viewerng/viewer?url=https://datafinder.ru/files/Cheat-Sheet-SUBD.pdf

Домашка №2 и №3

Интеграции

1. Взять ваши ФТ и НФТ
2. Подумать какие сервисы будут и какие интеграции между ними лучше сделать
3. Выбрать способ взаимодействия между сервисами, написать обоснование, сделать верхнеуровневые схемы

Базы данных

4. Взять ваши ФТ и НФТ
5. Подумать какие БД будут под какие сценарии, описать сценарий выбора
6. Описать дополнительно: репликация, шардинг

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ