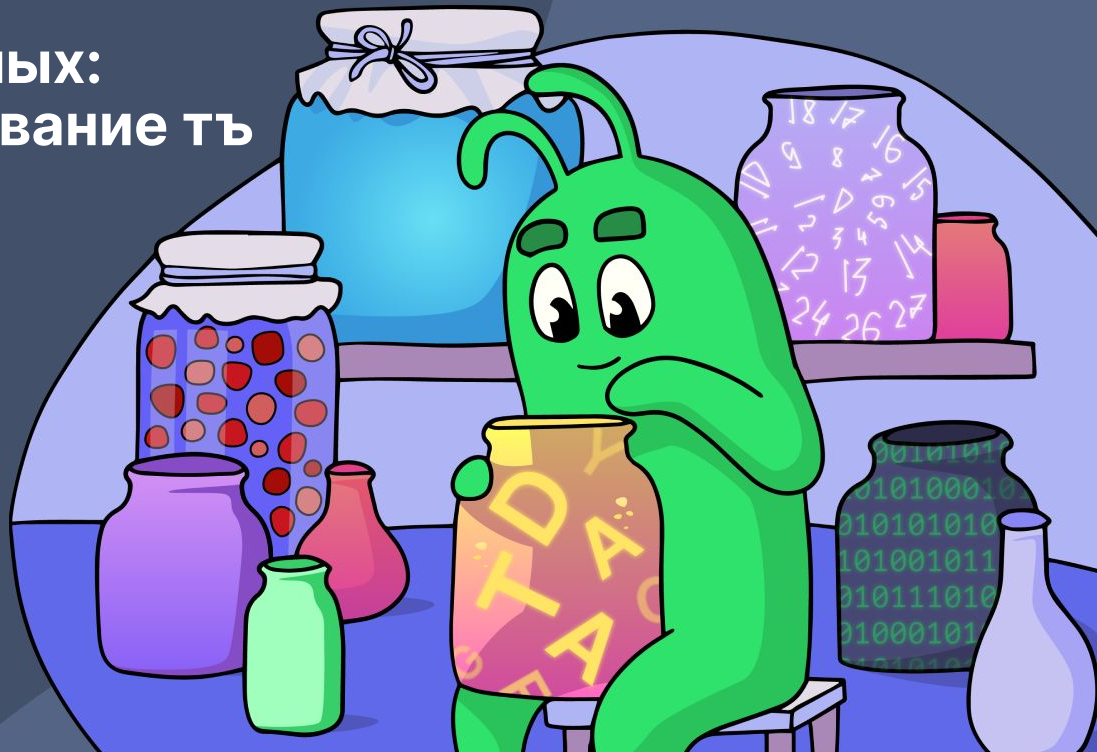
СЛЁРМ Обучение для инженеров и технических лидеров

Дата-инженер

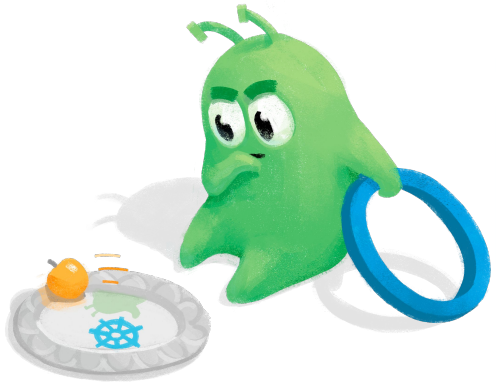
Реляционные базы данных: хранение и масштабирование тъ



Николай Марков

Цели урока. Что вы узнаете:

1. Как устроены реляционные БД?
2. Какова роль схем в базах данных?
3. Каким образом базы данных масштабируются ?
4. Что означают слова — «шардинг», «репликация», «консенсус»?



Суть реляционных БД

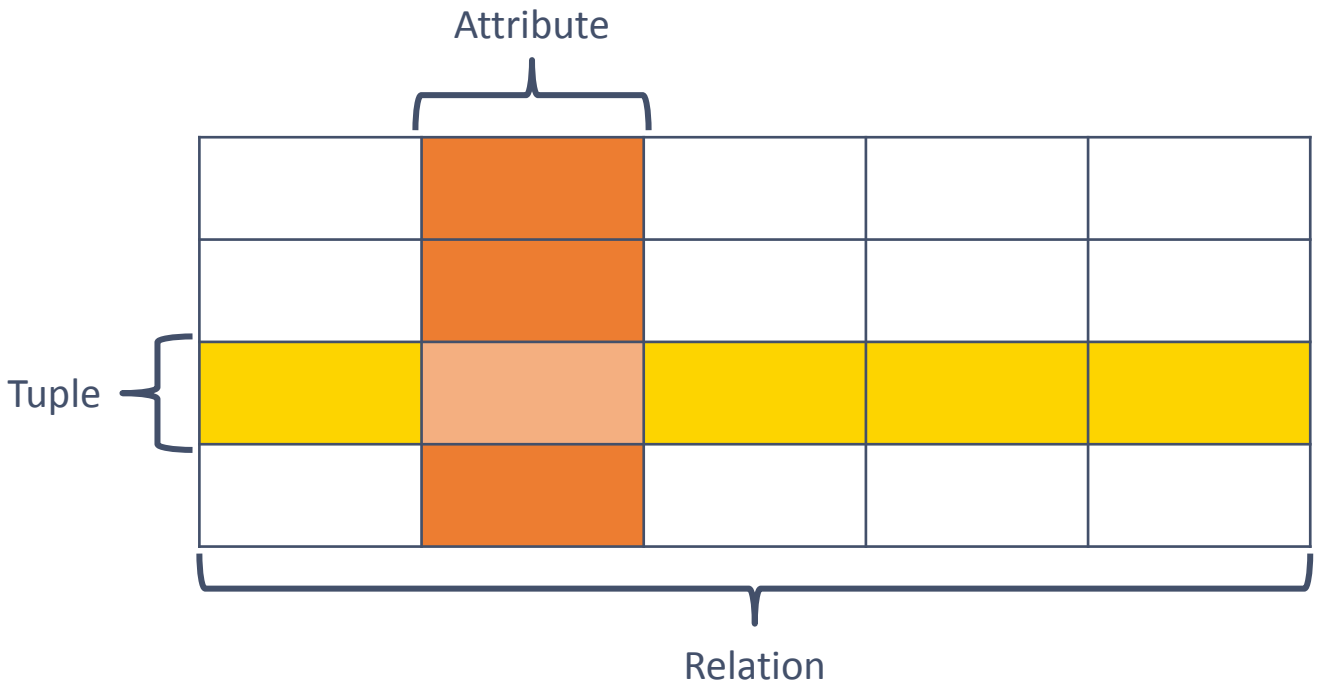
Терминология

Термины «база данных» и «система управления базами данных» обычно взаимозаменяемы

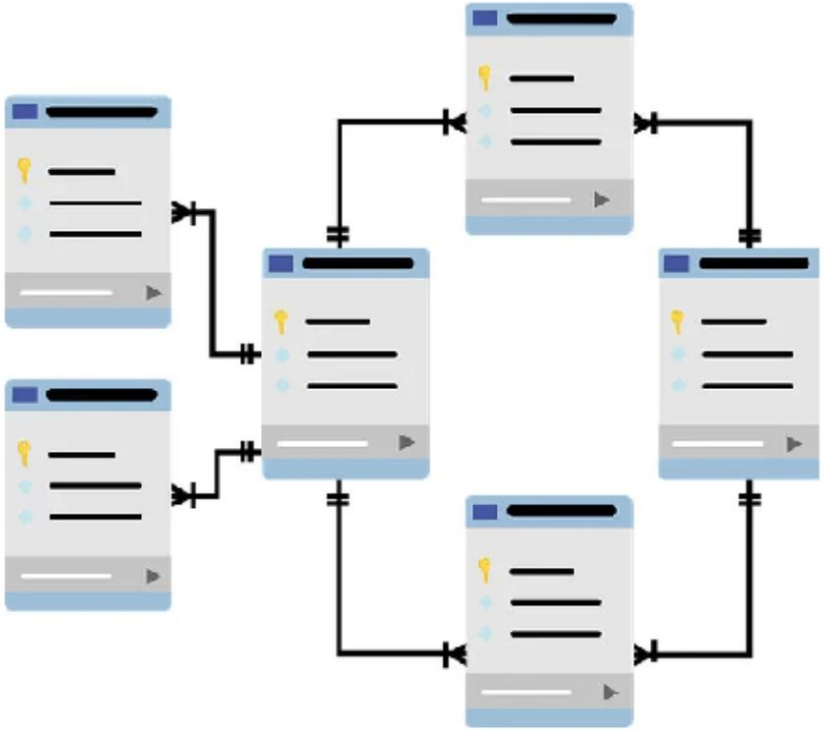
Реляционная БД не обязана поддерживать SQL, равно как и SQL работает не только поверх реляционных баз

«Реляционный» - от слова relation
и изначально описывает не связь между таблицами, а группу наборов полей в одной таблице

Relations



Связи

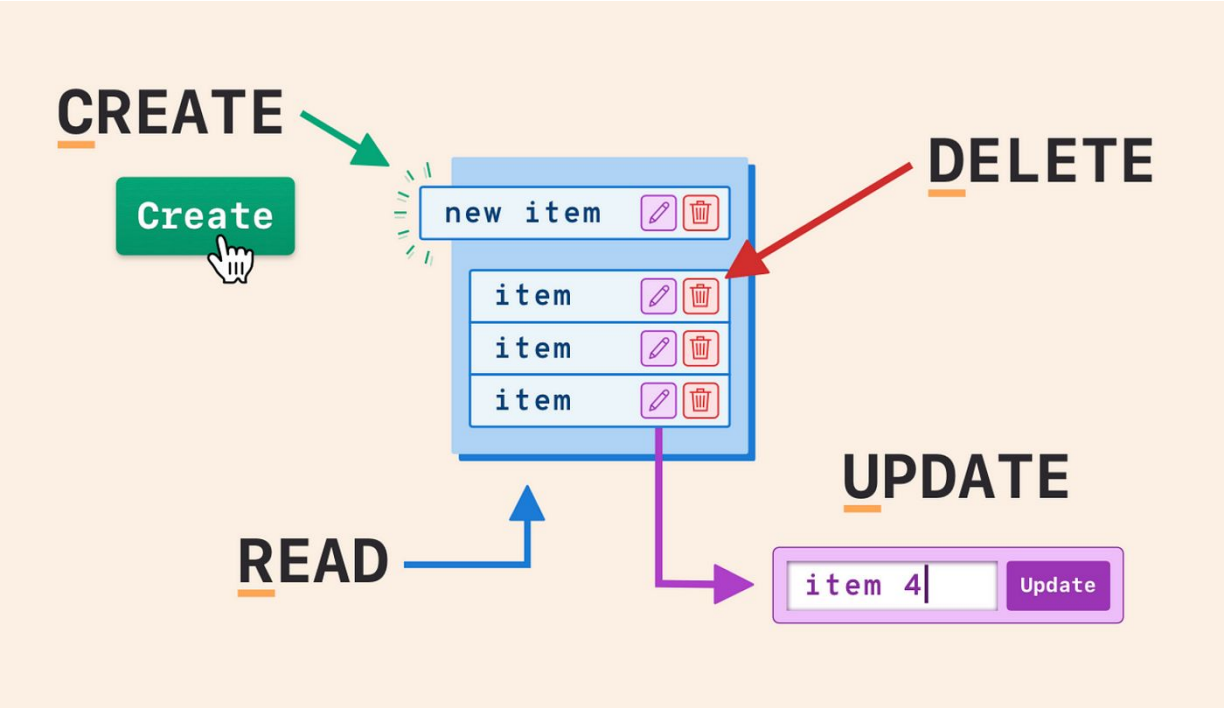


One-to-One

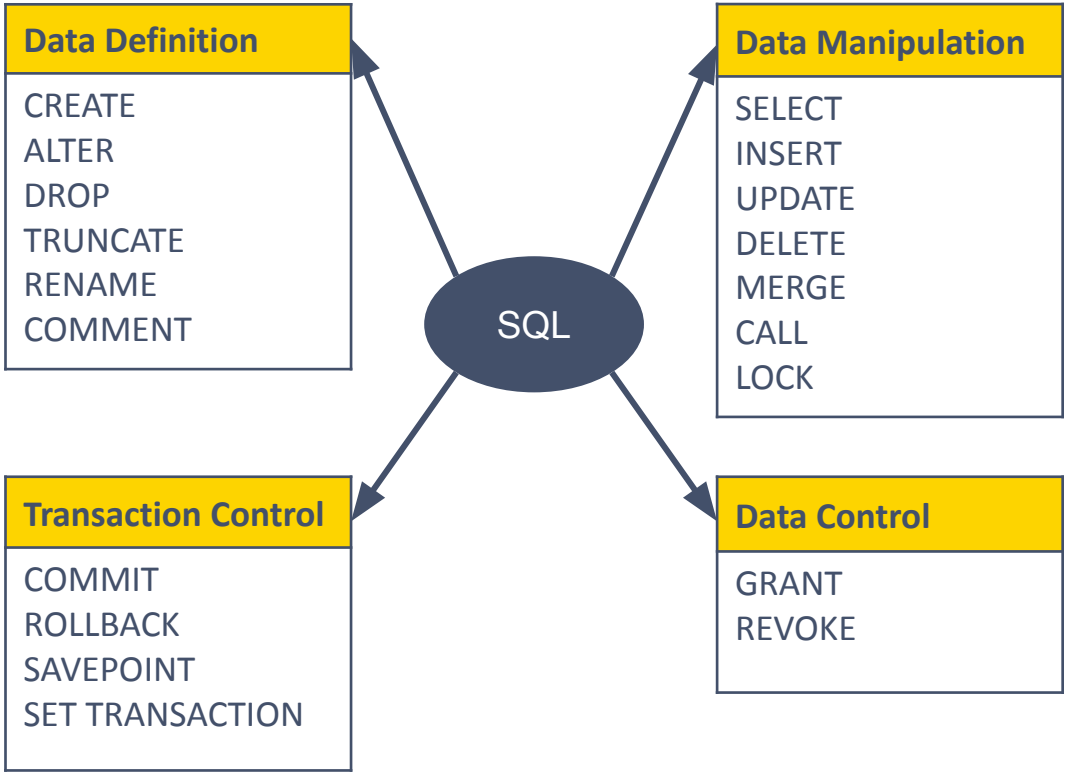
One-to-Many

Many-to-Many

Create/Read/Update/Delete



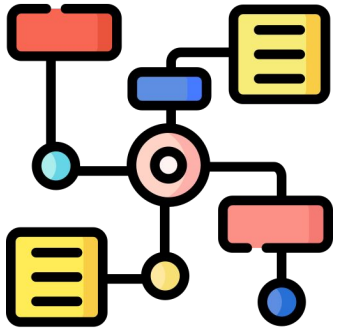
DDL/DML/DCL/TCL



Схемы хранения

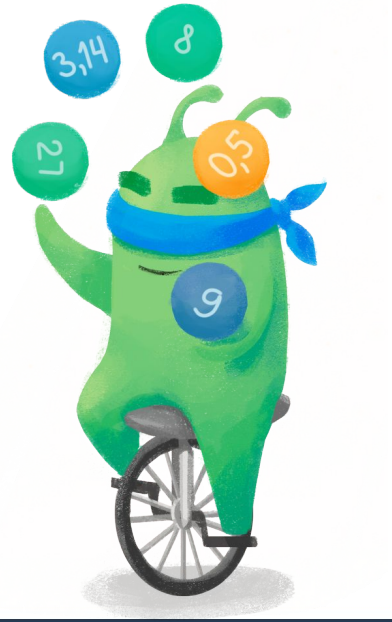
Schema on Read / Schema on Write

- **Реляционные базы** преимущественно используют **SoW**
- Типы колонок надо задать **при создании таблицы**,
  при заливке данных заранее **привести их к нужному типу**
- Другие виды баз могут быть «schemaless», т.е. **использовать SoR**
- Например, чтение данных из файла JSON из **объектного хранилища S3** — это процедура **SoR**

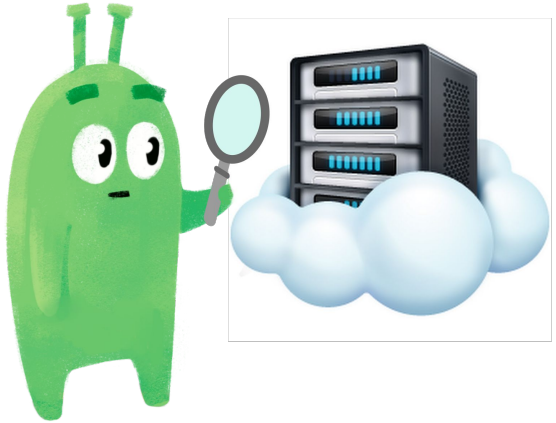


Типы данных в PostgreSQL

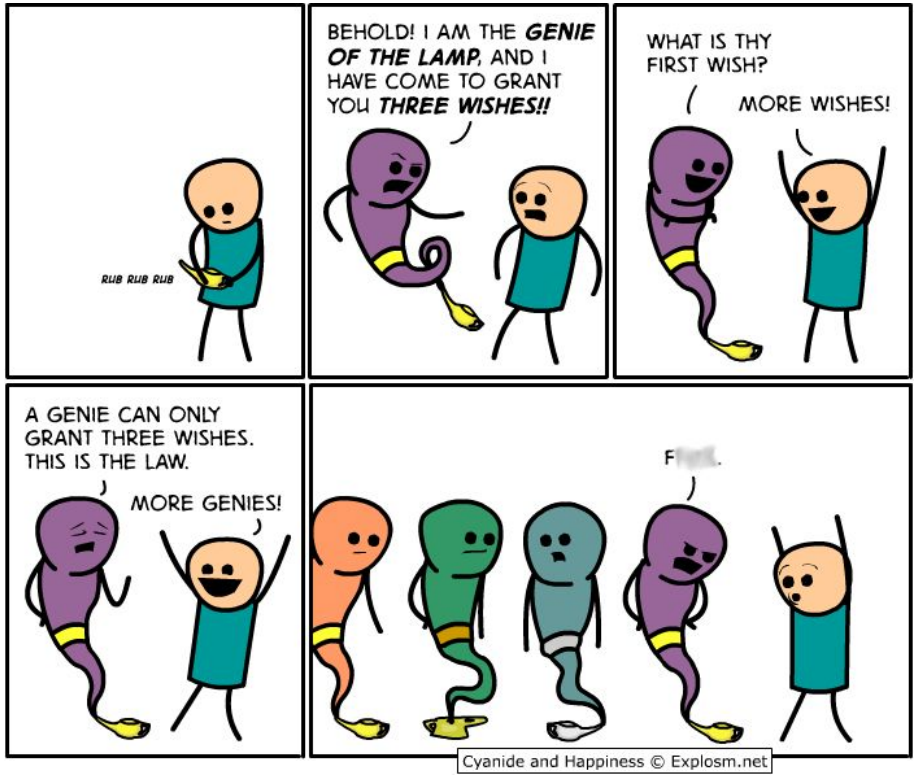
- **Текстовые** (CHAR, VARCHAR, TEXT)
- **Целочисленные** (INTEGER, SERIAL)
- **Фиксированной точности** (DECIMAL, NUMERIC)
- **С плавающей точкой** (REAL, DOUBLE)
- **Логические** (BOOL)
- **ВременнЫе** (DATE, TIME, TIMESTAMP)
- **Бинарные** (BYTEA, JSONB)



Как базы масштабируются



Вертикальное и горизонтальное масштабирование



Репликация

фактор репликации

Разбиваем данные на куски, делаем N копий каждого куска, каждую копию записываем на отдельную машину

Данные географически ближе к пользователям

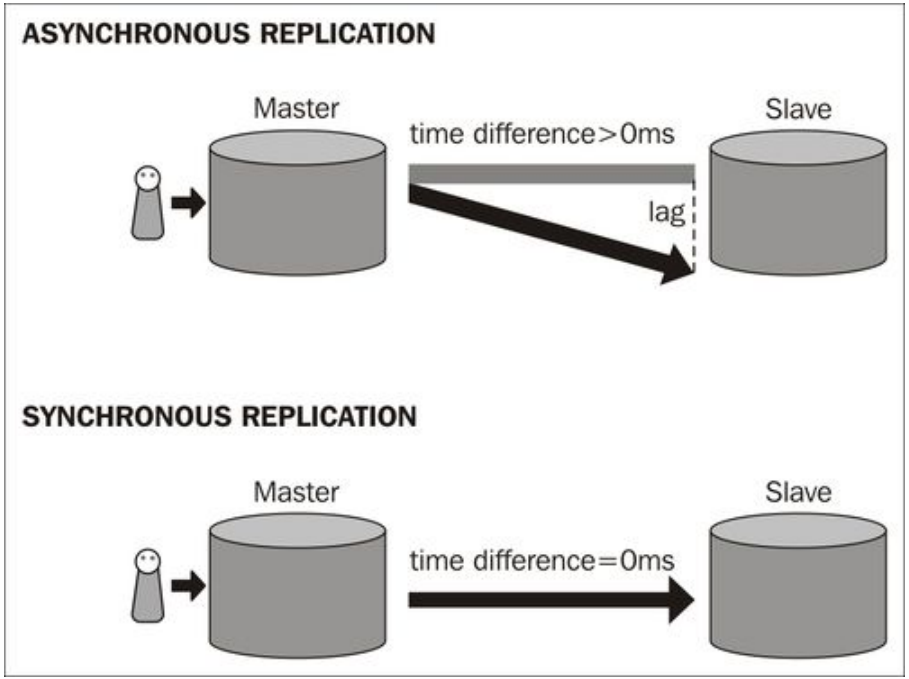
Выше Availability

Выше Throughput

Типы репликации

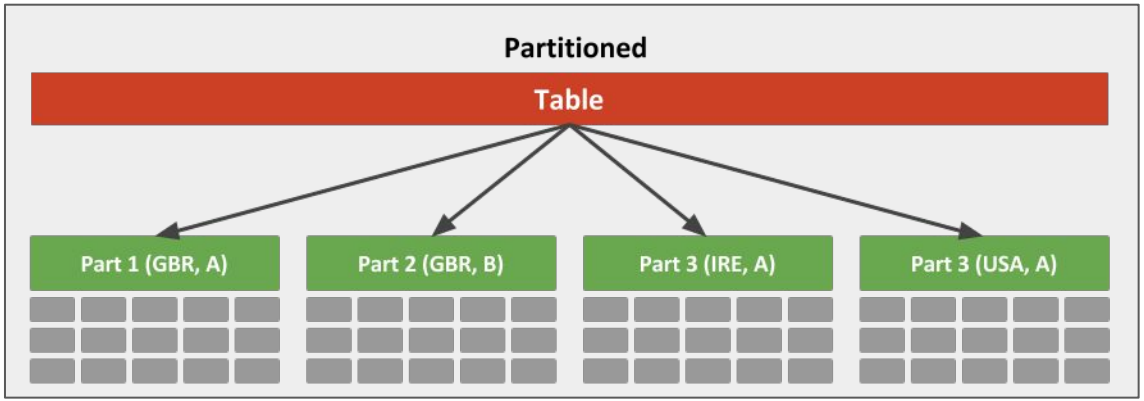
Leader -> Follower

Leader -> Leader



Партиционирование и шардинг

Хотим иметь несмещенные данные

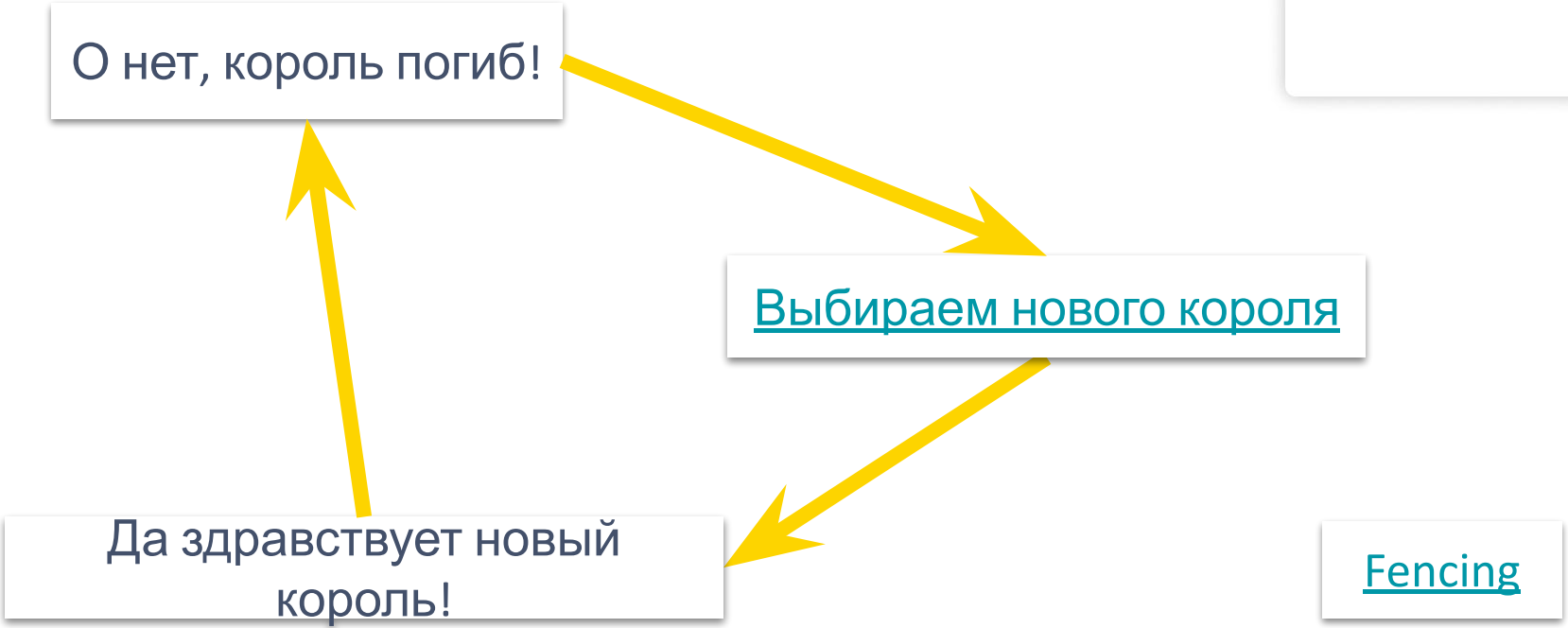


Key Ranges?

Hash of Key?

Consistent Hashing

О решении проблем при падении



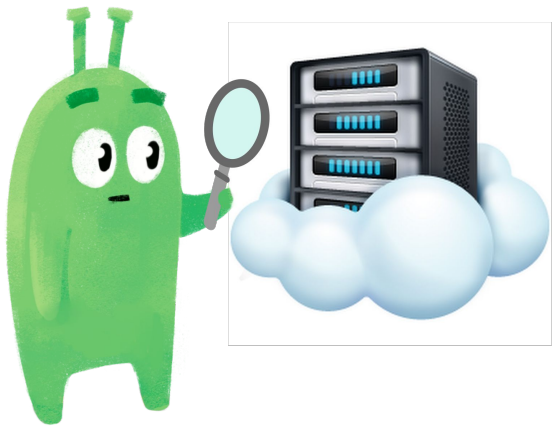
Траблшутинг

Мониторинг, а еще лучше observability - главный помощник в решении любых проблем с БД

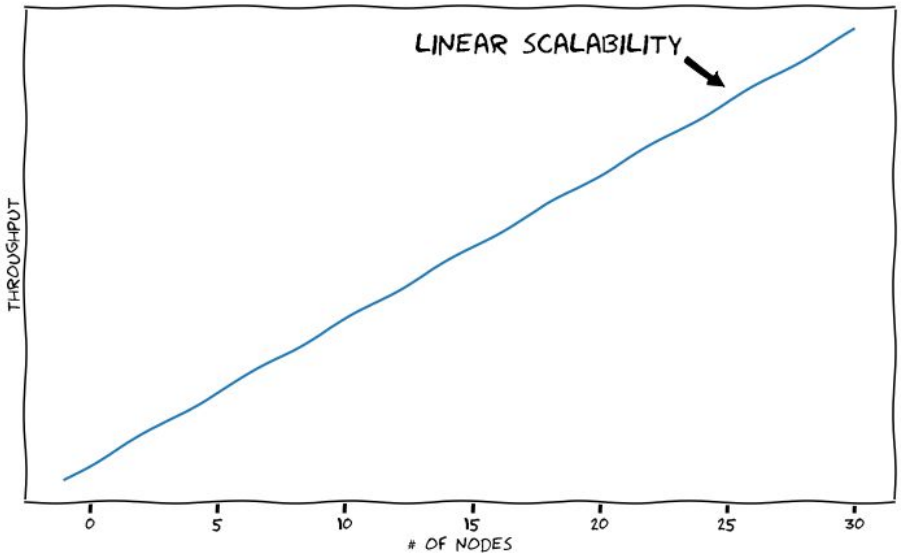
Одна замедленная или неконсистентная нода может сломать весь кластер

Далеко не факт, что вам потребуется настраивать все эти вещи руками, но умение разговаривать на одном языке с DBA может крайне пригодиться в боевых условиях

Чуть больше о распределенных системах



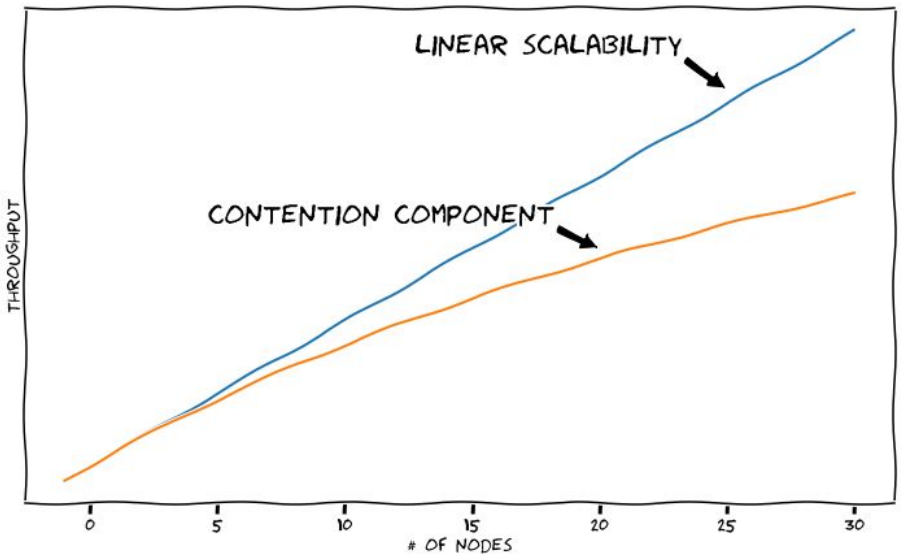
Линейное масштабирование



$$Throughput(N) = \frac{\lambda N}{1}$$

$\lambda -$ coefficient of performance

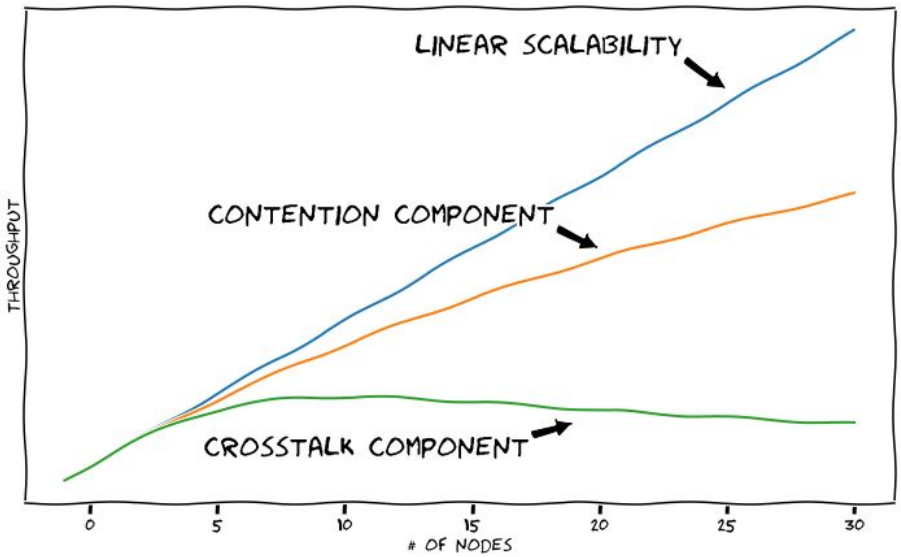
Contention component



$$Throughput(N) = \frac{\lambda N}{1+\sigma(N-1)}$$

σ – coefficient of serialization (contention)

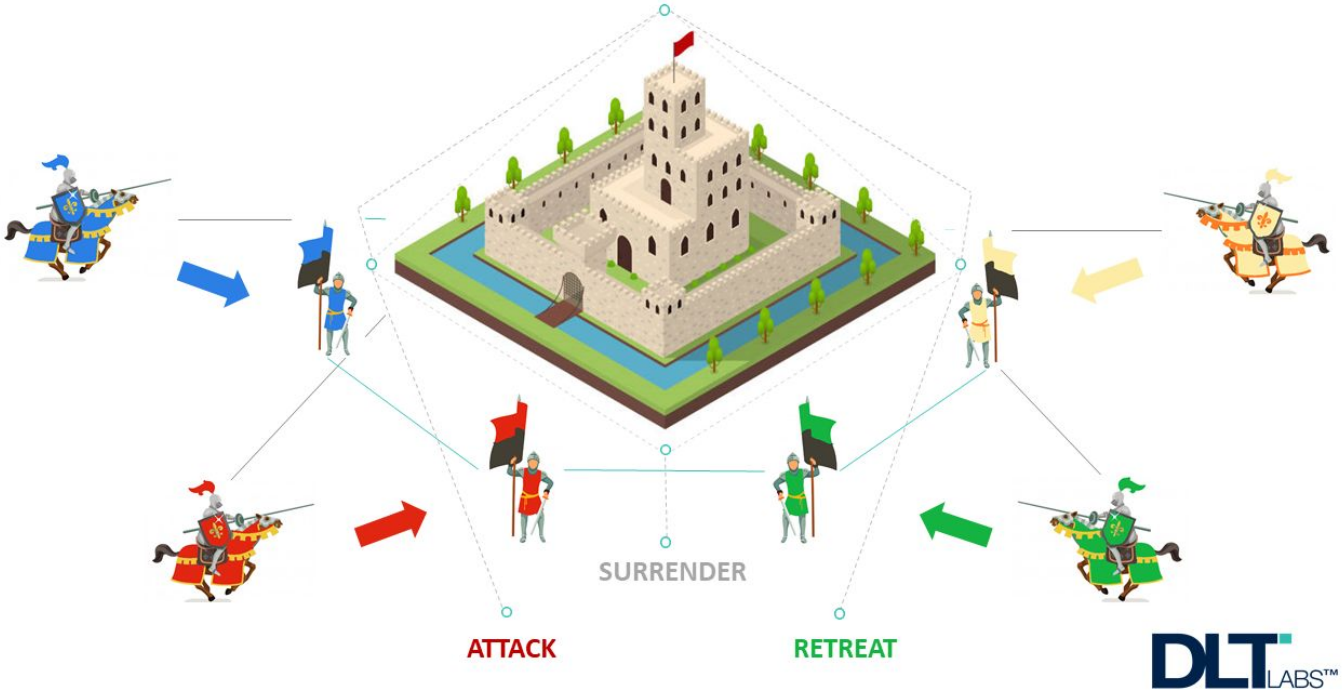
Crosstalk component



$$Throughput(N) = \frac{\lambda N}{1+\sigma(N-1)+kN(N-1)}$$

$k -$ crosstalk coefficient

Проблема византийских генералов



Консенсус и консистентность

MIT Course on Distributed Systems: Raft (1, 2)

Paxos, Raft, CRDT

Tendermint

Apache ZooKeeper™

etcd

Итоги. О чем поговорили:

1. Почему реляционные базы так называются
2. Что такое Schema-on-Read и Schema-on-Write
3. Как базы данных масштабируются
4. Зачем в базах данных консенсус

Спасибо
за внимание!