

Реляционные VS Нереляционные базы данных



Стариченко Никита

LinkedIn: www.linkedin.com/in/nikita-starichenko/
Telegram: @nikita_starichenko

6+ Years of Experience:

STO solutions, San Francisco USA

01/2020 - now

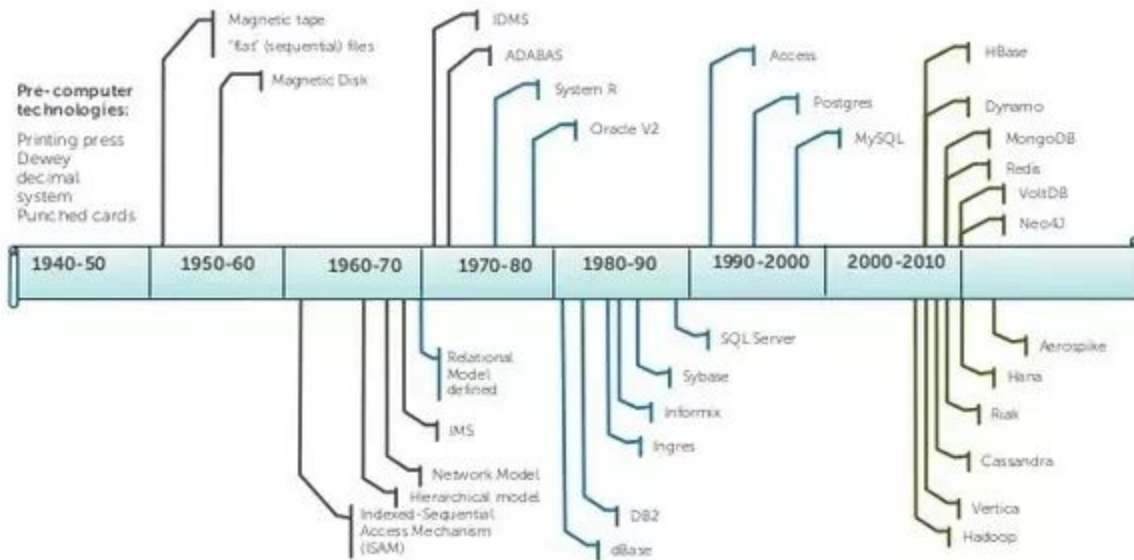
- Designing and Developing SPA application for healthcare risk adjustment automation
- Designed architecture of the entire system from scratch

Dodo Pizza, Oxford USA / Moscow Russia

03/2018 – 04/2019

- Developing and maintaining DODO IS
- Reduced release time by 40% by fixing more than 50 UI tests that led to a decrease of manual testing
- Piloted first microservice on .Net Core and GRPC that is composing by Docker that uses a full CI/CD including integration tests and run in Kubernetes
- Piloted integration React to Angular.js and add ability to step by step rewrite frontend from Angular to React that speeded up front development by 2 times
- Got rid of the need to restart the system by eliminating the daily memory leak of 100mb by finding that leak in .Net Core application using memory snapshot tools on Linux in runtime

History of databases



Зачем так много?

1. Хранение данных
2. Больших-Маленьких
3. Долго-Мало
4. Однообразных-Разных
5. Важных-Не очень
6. OLAP-OLTP
7. Сессии
8. Поиск
9. Отношения



Обзор

Реляционные:

1. Структура/Схема
2. Транзакции

Примеры

1. Oracle
2. MySQL
3. SQL Server

Нереляционные:

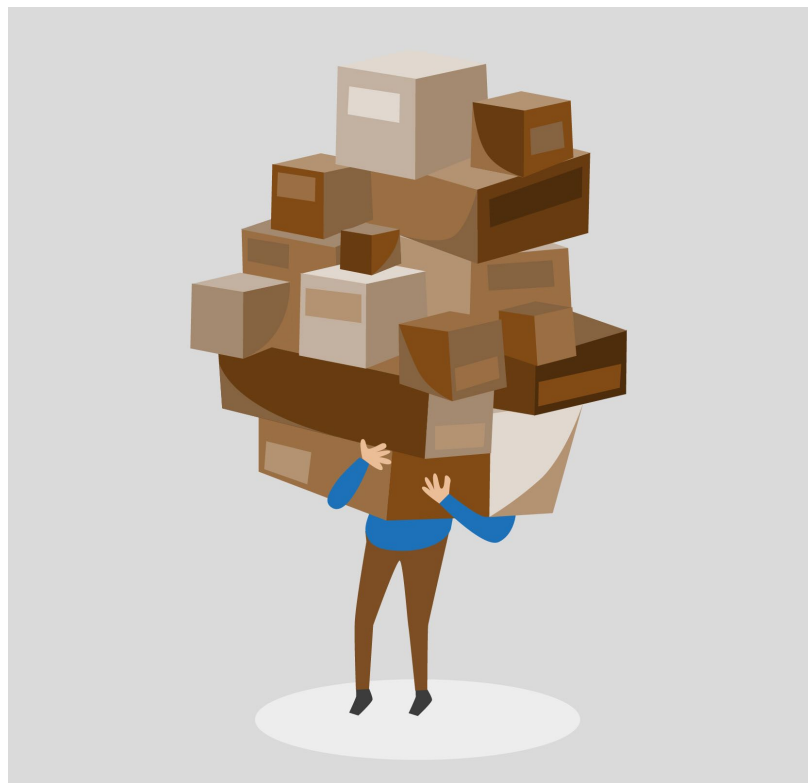
1. Динамика

Примеры

1. mongoDB
2. redis
3. cassandra

Реляционность

1. Нормализация
2. SQL
3. Целостность данных
4. Транзакции (ACID)



Разновидности баз

1. Реляционная
2. Ключ-значение
3. Документная
4. Графовая
5. Поисковая
6. МногоСтолбцовая
7. Временные ряды
8. Многомодельная

1. SQL
2. Redis
3. MongoDB
4. Neo4j
5. Elasticsearch
6. Cassandra
7. InfluxDB
8. Couchbase

Рабочие нагрузки

Реляционные:

1. OLTP
2. OLAP

Нереляционные:

1. Разные виды доступа
2. Разное время отклика
3. Аналитика динамических данных



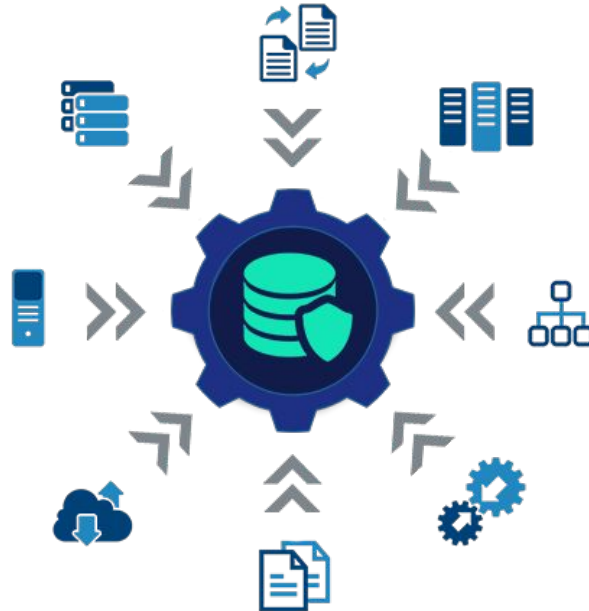
Модель данных

Реляционные:

1. Таблицы-строки-столбцы
2. Схема
3. Нормализация

Нереляционные:

1. Ключ-значение
2. Документы
3. Графы
4. Временные ряды
5. Поисковая



Транзакции

Реляционные:

1. ACID
2. Атомарность
3. Непротиворечивость
4. Изолированность
5. Надежность

Нереляционные:

1. BASE
2. В основном доступно
3. Мягкое состояние
4. Конечная согласованность



Производительность

Реляционные:

1. Оптимизация
2. Запросы
3. Индексы
4. Структура

Нереляционные:

1. Конкретный вид базы



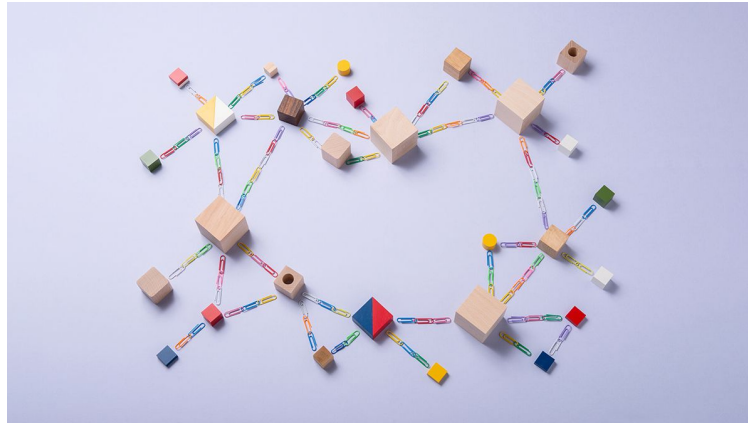
Масштабирование

Реляционные:

1. Вертикальное
2. Горизонтальное для чтения

Нереляционные:

1. Горизонтальное



SQL

Oracle:

1. Документ, Граф, RDF
2. Все
3. Oracle
4. Платная

MySQL:

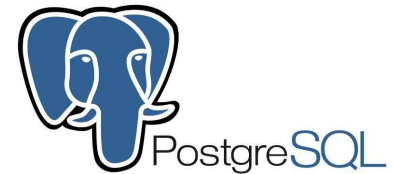
1. Документ
2. Все
3. Oracle
4. Бесплатная

SQL Server:

1. Документ, Граф
2. Linux / Windows
3. Microsoft
4. Бесплатная, Платная

PostgreSQL:

1. Документ
2. Все
3. PostgreSQL
4. Бесплатная



mongoDB vs SQL

Отличия:

1. Документная БД
2. Нет схемы



mongoDB

Плюсы:

1. Шардинг из коробки (> вместимость)
2. Нагрузка (> запросов в секунду)
3. Большая скорость записи

Минусы:

1. Большая нагрузка на CPU (пример: join)

redis vs SQL

Отличия:

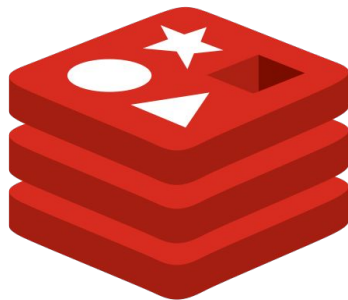
1. Ключ-Значение
2. Нет схемы
3. В памяти
4. Строки, списки, множества
5. Нет языка запросов

Плюсы:

1. Шардинг из коробки (по хэшу)
2. Очень быстрое чтение/запись

Минусы:

1. Большая нагрузка на оперативную память



redis

elasticsearch vs SQL

Отличия:

1. Поисковая БД
2. Нет схемы
3. JSON

Плюсы:

1. Шардинг из коробки
2. Очень быстрый поиск данных

Минусы:

1. Нет транзакций



elasticsearch

cassandra vs SQL

Отличия:

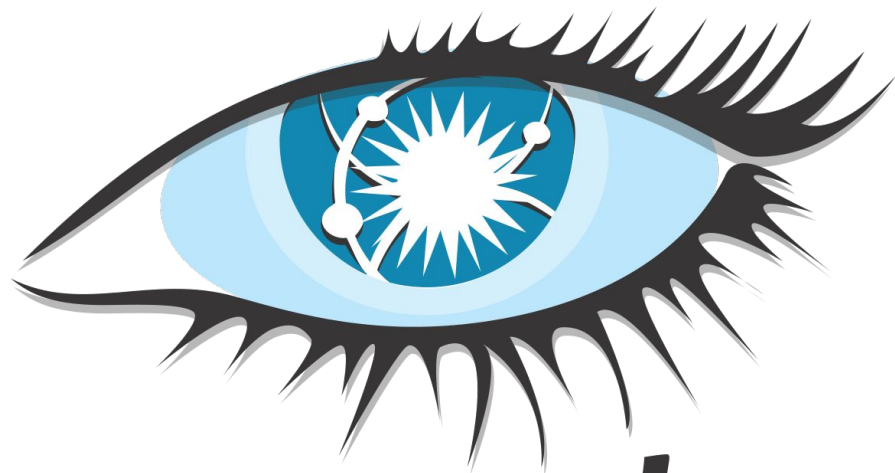
1. Многостолбцовая БД
2. Строки с полями
3. CQL

Плюсы:

1. Шардинг из коробки (по хэшу)
2. Быстрая запись
3. Быстрое чтение по ключу

Минусы:

1. Нет полноценных транзакций
2. Медленные агрегатные запросы



cassandra

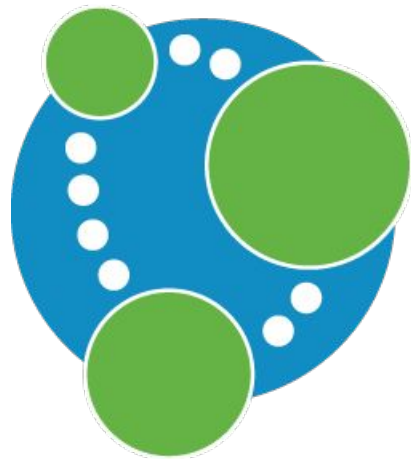
neo4j vs SQL

Отличия:

1. Графовая БД
2. Узлы, Отношения
3. Нет четкой схемы

Плюсы:

1. Шардинг
2. Транзакции
3. Быстрый поиск в глубину
4. Простой поиск в глубину



neo4j

influxdb vs SQL

Отличия:

1. Временные Ряды
2. Есть схема

Плюсы:

1. Шардинг
2. Очень быстрая запись
3. InfluxQL похож на SQL

Минусы:

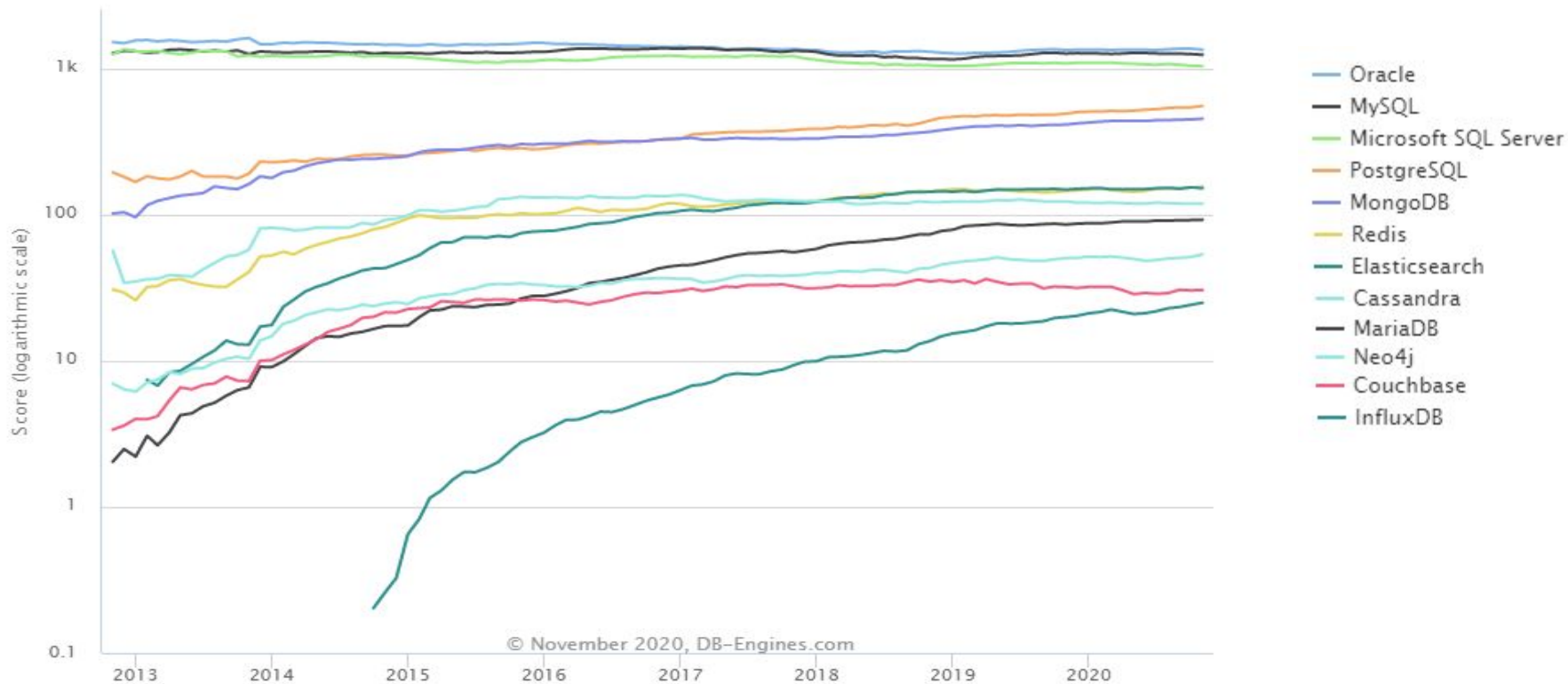
1. Нет обычного удаления



influxdb

Популярность

DB-Engines Ranking



Что использовать?

Реляционные:

1. Структурированные данные
2. ACID

Нереляционные:

1. Гибкость
2. Скорость
3. Масштабируемость
4. Другое



Итог:

1. Баз очень много
2. Областей применения ещё больше
3. Можно скрещивать подходы
4. Выбирайте головой

Ресурсы:

1. <https://db-engines.com/>
2. Официальные сайты каждой базы данных