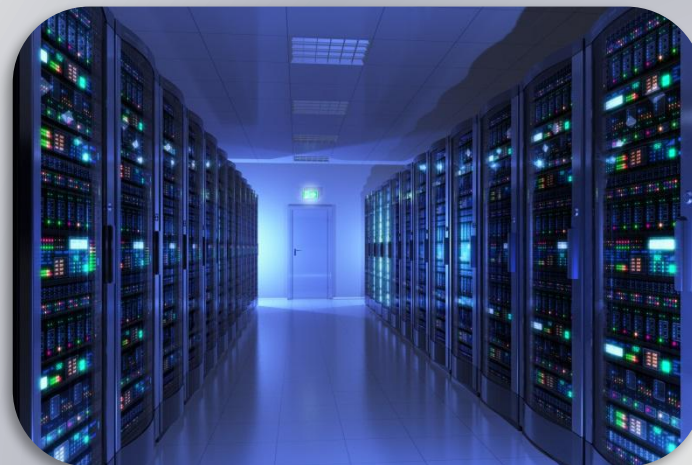


A conceptual image featuring a globe constructed from interlocking puzzle pieces. The globe is positioned on the left side of the frame, with several individual puzzle pieces scattered on the light-colored background to its right. The text 'SQL u NoSQL' is overlaid in the center of the image in a white, italicized, sans-serif font. The 'u' is smaller and positioned between 'SQL' and 'NoSQL'.

*SQL u NoSQL*

# Вступление

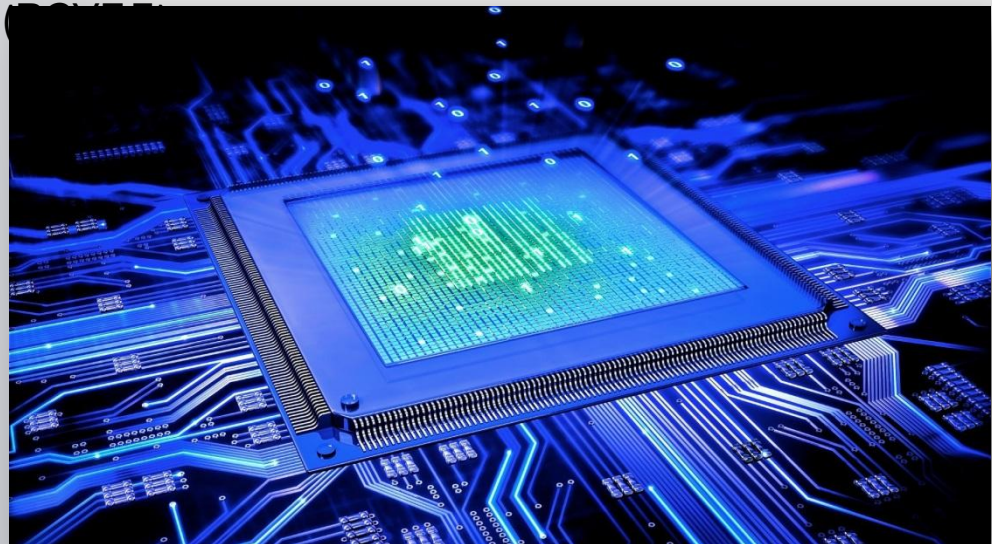
С незапамятных времен память была одной из самых важных и необходимых составляющих компьютера. Несмотря на разницу в методах реализации, большинство вычислительных машин оснащены необходимым аппаратным обеспечением для обработки и хранения информации. В наше время невозможно представить работу какого-либо приложения, хоть игры, хоть сайта, без получения, обработки и записи определённого типа данных. **Системы управления базами данных (СУБД)** — это высокоуровневое программное обеспечение, работающее с низкоуровневыми API. Для решения различных проблем создавались новые виды СУБД (реляционные, NoSQL и т.д.) и их новые реализации (MySQL, PostgreSQL, MongoDB, Redis и т.д.). В этой статье мы разберемся в основах баз данных и СУБД.



# Системы управления базами данных (СУБД)

СУБД — это общий термин, относящийся ко всем видам абсолютно разных инструментов, от компьютерных программ до встроенных библиотек. Эти приложения управляют или помогают управлять наборами данных. Так как эти данные могут быть разного формата и размера, были созданы разные виды СУБД. СУБД основаны на **моделях баз данных** — определённых структурах для обработки данных. Каждая СУБД создана для работы с одной из них с учётом особенностей операций над информацией.

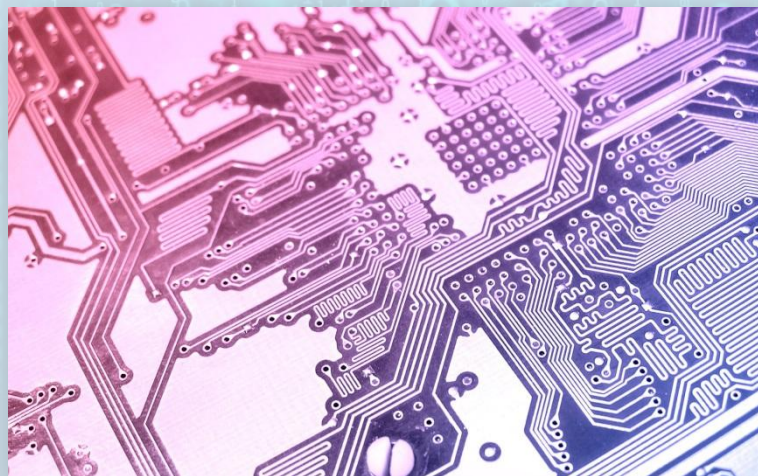
Хотя решений, реализующих различные модели баз данных, очень много, периодически некоторые из них становятся очень популярными и используются на протяжении многих лет. Сейчас самой популярной моделью является **реляционная система управления базами данных**



# Модели баз данных

Каждая СУБД реализует одну из моделей баз данных для логической структуризации используемых данных. Эти модели являются главным критерием того, как будет работать и управлять информацией приложение. Существует несколько таких моделей, среди которых самой популярной является реляционная.

Хотя она и является весьма мощной и гибкой, есть ситуации, решения которых она предложить не может. Тут на помощь придёт сравнительно новая модель, называемая NoSQL. Она набирает популярность и предлагает весьма интересные решения и дополнительный функционал. Из-за того, что эти системы не используют строгую структуризацию данных, они предлагают большую свободу действий при обработке информации.

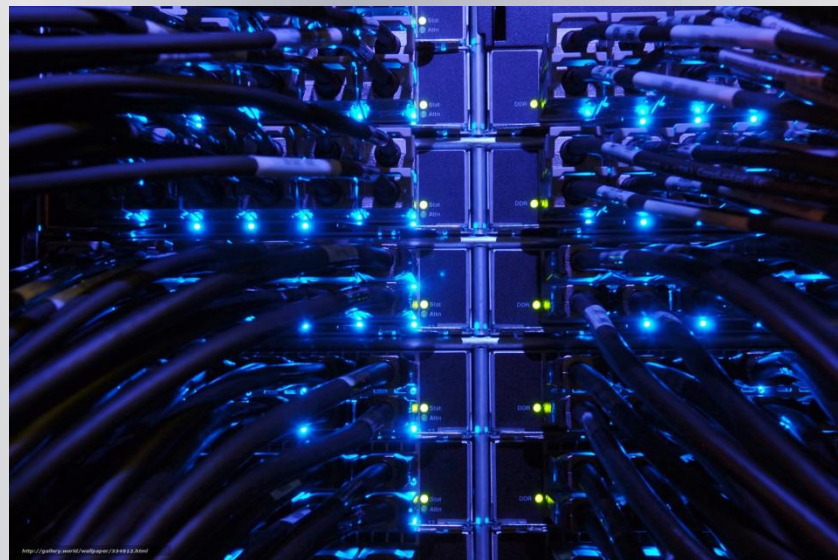
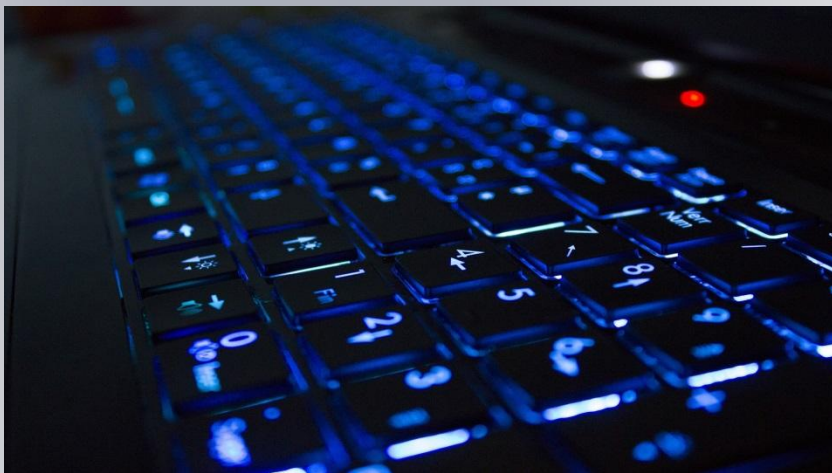


# Реляционная модель

Представленная в 70-х, реляционная модель предлагает математический способ структуризации, хранения и использования данных. Отношения (англ. *relations*) дают возможность группировки данных как связанных наборов, представленных в виде таблиц, содержащих упорядоченную информацию (например, имя и адрес человека) и соотносящихся значения и атрибуты (его номер паспорта).

Благодаря десятилетиям исследований и разработки РСУБД работают производительнее и надёжнее. В сочетании с большим опытом использования администраторами реляционные базы данных стали выбором, гарантирующим защиту информации от потерь.

Несмотря на строгие принципы формирования и обработки данных, РСУБД могут быть весьма гибкими, если приложить немного усилий.



# Безмодельный (NoSQL) подход

NoSQL-способ структуризации данных заключается в избавлении от ограничений при хранении и использовании информации. Базы данных NoSQL, используя неструктурированный подход, предлагают много эффективных способов обработки данных в отдельных случаях (например, при работе с хранилищем текстовых документов).

## Популярные СУБД

### РСУБД

Реляционные системы управления базами данных берут своё название от реализуемой модели — реляционной. Сейчас они остаются, да и ещё какое-то время будут, самым популярным выбором для надёжного, безопасного и производительного хранения данных. РСУБД требуют чётких и ясных схем — не стоит путать со специфическим определением для PostgreSQL — для работы с данными. Эти рамки, определённые пользователем, задают способ их хранения и использования. Схемы очень похожи на таблицы, столбцы которых отражают порядковый номер и тип информации в каждой записи, а строки — содержимое этих записей.

Самыми популярными РСУБД сейчас являются:

**SQLite:** очень мощная встраиваемая РСУБД.

**MySQL:** самая популярная и часто используемая РСУБД.

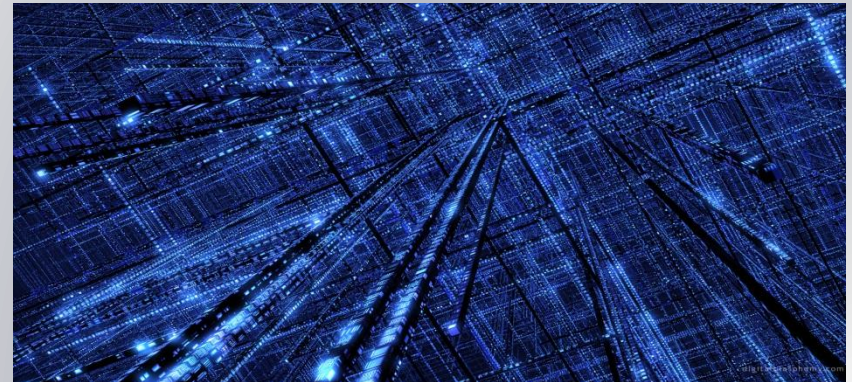
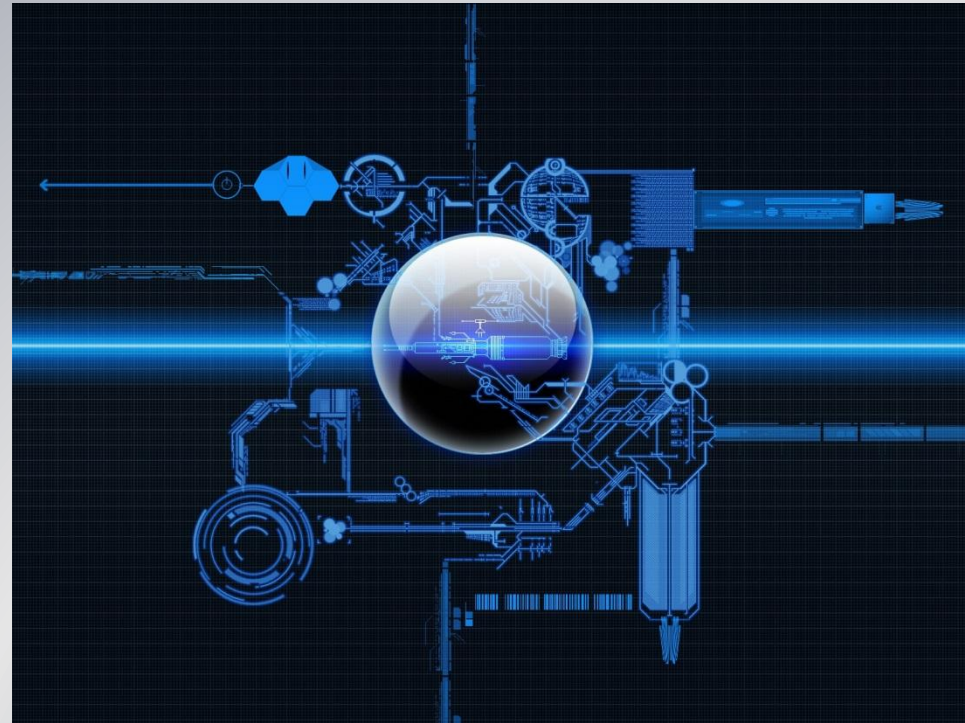
**PostgreSQL:** самая продвинутая и гибкая РСУБД.

# NoSQL-СУБД

NoSQL-СУБД не используют реляционную модель структуризации данных. Существует много реализаций, решающих этот вопрос по-своему, зачастую весьма специфично. Эти бессхемные решения допускают неограниченное формирование записей и хранение данных в виде ключ-значение.

В отличие от традиционных РСУБД, некоторые базы данных NoSQL, например, MongoDB, позволяют группировать коллекции данных с другими базами данных. Такие СУБД хранят данные как одно целое. Эти данные могут представлять собой одиночный объект наподобие JSON и вместе с тем корректно отвечать на запросы к полям.

NoSQL базы данных не используют общий формат запроса (как SQL в реляционных базах данных). Каждое решение использует собственную систему запросов.



# Сравнение SQL и NoSQL

Для того, что прийти к простому и понятному выводу, давайте проанализируем разницу между SQL- и NoSQL-подходами:

**Структура и тип хранящихся данных:** SQL/реляционные базы данных требуют наличия однозначно определённой структуры хранения данных, а NoSQL базы данных таких ограничений не ставят.

**Запросы:** вне зависимости от лицензии, РСУБД реализуют SQL-стандарты, поэтому из них можно получать данные при помощи языка SQL. Каждая NoSQL база данных реализует свой способ работы с данными.

**Масштабируемость:** оба решения легко растягиваются вертикально (например, путём увеличения системных ресурсов). Тем не менее, из-за своей современности, решения NoSQL обычно предоставляют более простые способы горизонтального масштабирования (например, создания кластера из нескольких машин).

**Надёжность:** когда речь заходит о надёжности, SQL базы данных однозначно впереди.

**Поддержка:** РСУБД имеют очень долгую историю. Они очень популярны, и поэтому получить поддержку, платную или нет, очень легко. Поэтому, при необходимости, решить проблемы с ними гораздо проще, чем с NoSQL, особенно если проблема сложна по своей природе (например, при работе с MongoDB).

**Хранения и доступ к сложным структурам данных:** по своей природе реляционные базы данных предполагают работу с сложными ситуациями, поэтому и здесь они превосходят NoSQL-решения.



# *Список источников*

<https://tproger.ru/translations/sql-nosql-database-models/>

