

Введение в базы данных

Обзор баз данных

Skillbox

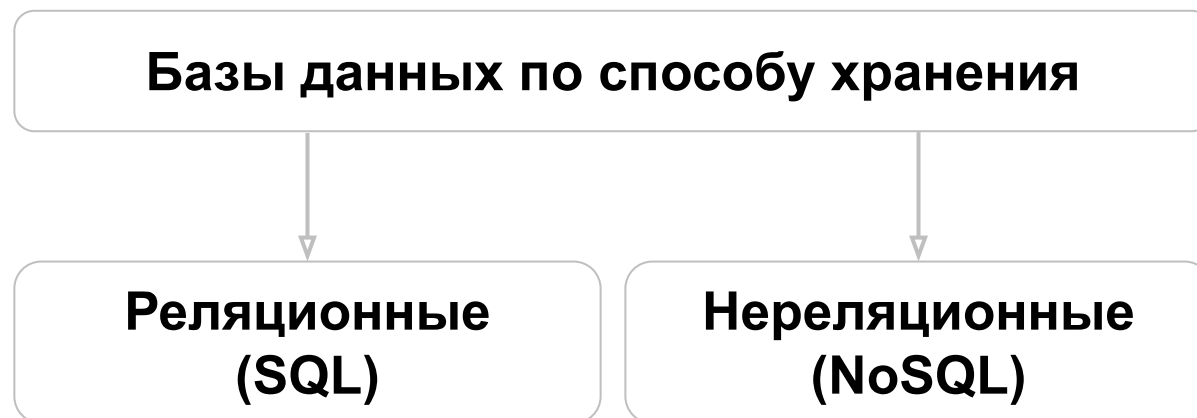
образовательная платформа

Типы БД

Это разделение связано со способом хранения информации.

Зависит от:

- предметной области
- способа обработки информации



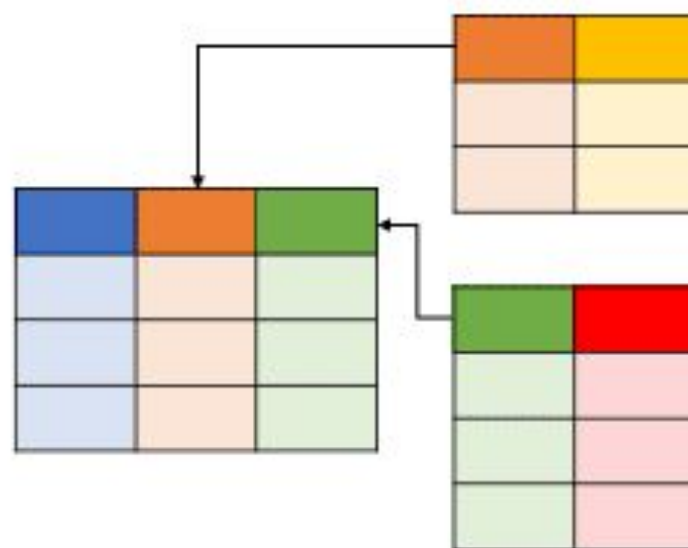
Реляционные БД (SQL)

Система, где данные хранятся в формате таблиц (отношений). Они строго структурированы и могут быть связаны друг с другом.

В таблице есть:

- строки = отдельная запись
- столбцы = поле с назначенным типом данных
- ячейки со своим типом данных

SQL



Реляционные СУБД

Каждая таблица в базе имеет жёсткую схему данных и не может быть нарушена

Пример SQL БД

Первая строка — это название колонок. Каждая строка описывает конкретную покупку клиента.

ID клиента	Стоимость заказа	Дата покупки
1	1 300	2021-12-12
2	2 900	2021-12-15
3	800	2021-11-01

Преимущества и недостатки

Плюсы



- Надёжность и неизменяемость данных
- Низкий риск потери информации
- Гарантия целостности данных независимо от ошибок или обновлений версий БД

Выполнение набора требований ACID:

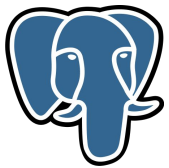
- Atomicity — Атомарность
- Consistency — Согласованность
- Isolation — Изолированность
- Durability — Прочность

Минусы

- Низкая скорость работы
- Сложности при горизонтальном масштабировании

Примеры реляционных баз

Самые известные производители ПО:



PostgreSql



Microsoft SQL



Oracle Database

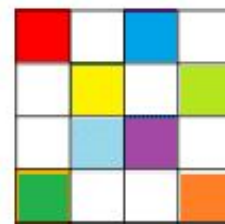
Нереляционные БД (NoSQL)

Система, где данные хранятся без чётких связей друг с другом и чёткой структуры.

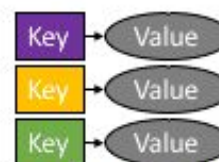
Вместо таблиц, внутри базы находится:

- множество разнородных документов
- блоков данных (изображения, видео или скрипт)

NoSQL



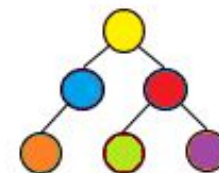
Колоночные



Ключ-значение



Графовые



Документо-ориентированные

Преимущества и недостатки

Достоинства

- Высокая скорость при слабоструктурированной информации
- Хорошая масштабируемость

Недостатки

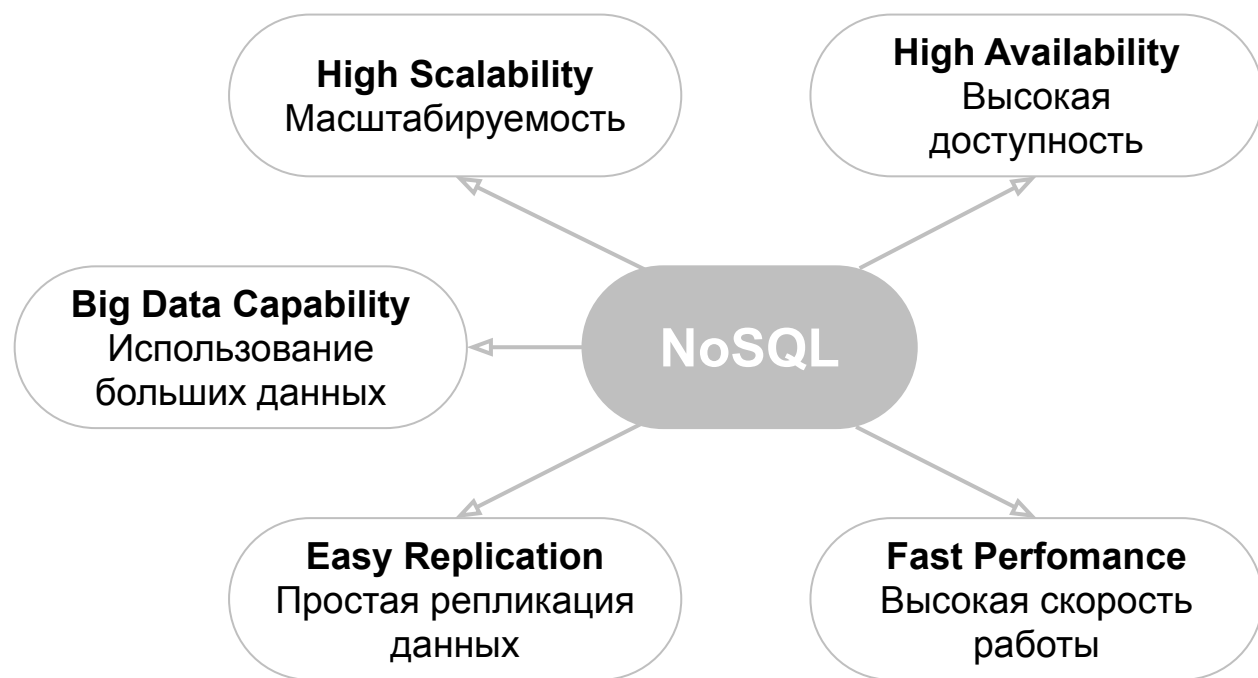
- Низкие требования к обеспечению целостности
- Ограничения транзакционности

Виды NoSQL БД

Деление всегда обусловлено особенностями применения: т. е. какую именно задачу БД лучше всего выполняет, под это её и используют.

Виды нереляционных БД:

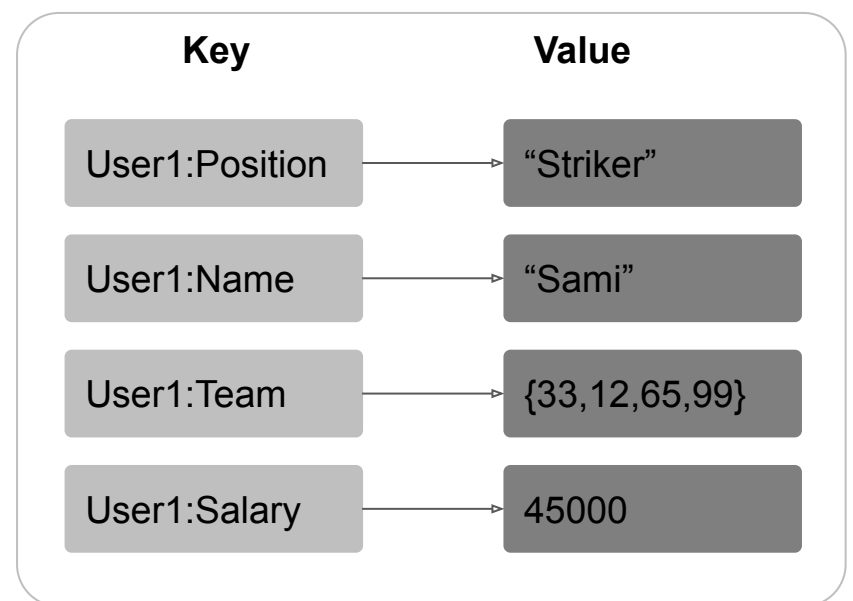
- Key-value
- документная
- графовая
- колоночные
- временных рядов



Key-Value БД

Переводится как «Ключ-Значение», данные хранятся по принципу словаря.

- Подходит для простых структурированных данных
- Часто используют для быстрого доступа (кеш)



Принцип работы

Любые значения (value) ассоциированы с ключом (key), с которым происходила их запись.

↓

С помощью этого ключа можно получить или перезаписать значения в будущем.

Применение Key-Value БД

Подходит для атрибутов с простой структурой и необходимостью быстрой работы.

Например:

- организация кеша простых данных
- хранение учётных записей пользователей

Популярные Key-Value БД



redis

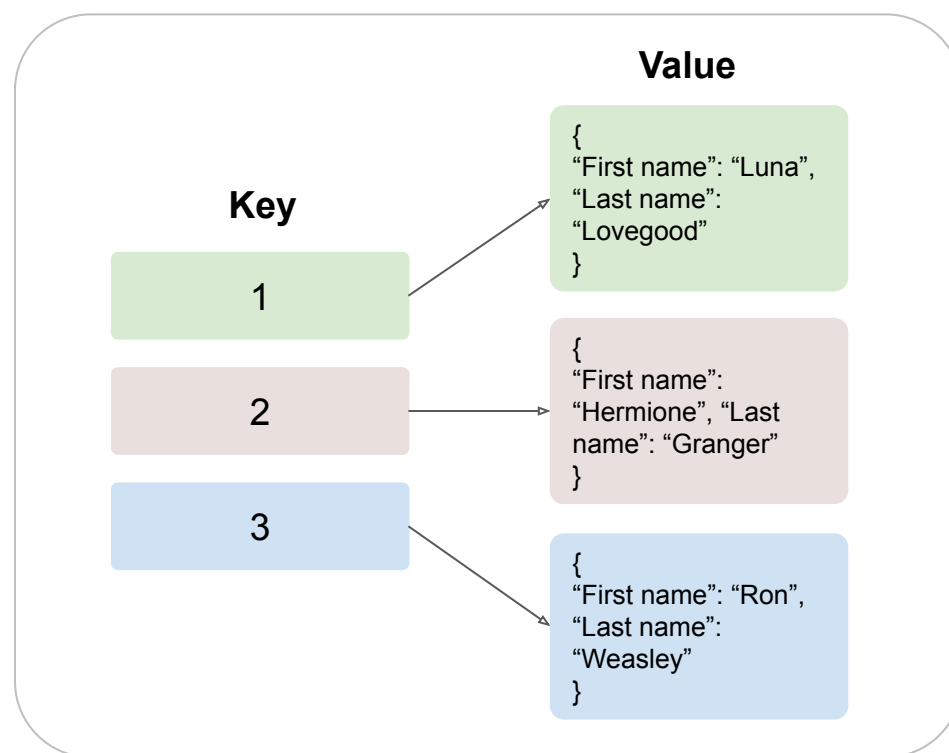


Документные БД

Данные хранятся в виде документов в формате JSON, XML или BSON, а внутренняя структура документов не зависит от других.



Документы можно вкладывать друг в друга, что создаёт между ними иерархическую связь.



Принцип работы

Все данные находятся в одном документе. Чтобы добавить новые, достаточно создать в документе дополнительное поле.

Применение документных БД

Подходит для:

- слабоструктурированных данных и высоких требований к скорости работы
- задач по уменьшению работ по реализации логики, связанной с БД и её структурой

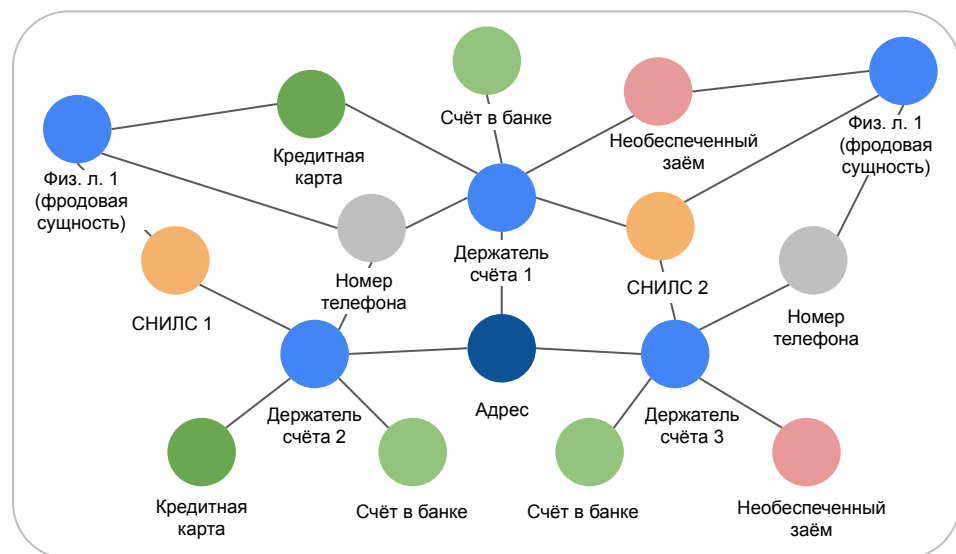
Популярные документные БД



Графовые БД

Данные состоят из узлов и связей между ними.

Узлы — это элементы со своими значениями. Они могут быть связаны между собой, что приводит к образованию сети взаимосвязей.



Принцип работы

БД работает, ориентируясь на связи между узлами, и эти связи могут иметь разные характеристики.

Здесь нельзя использовать язык запросов SQL

Применение графовых БД

Этот тип БД лучше всего использовать, когда приоритетными являются различные взаимосвязи между данными.

Популярные графовые БД



Колоночные БД

Данные группируются не по строкам, а по столбцам.

Пример

В реляционной БД:

- 1. 001, Vladimir, Ivanov, 34
- 2. 002, Dmitry, Sidorov, 56
- 3. 003, Aleksandr, Petrov, 43

В колоночной БД:

- 1. 001, 002, 003
- 2. Vladimir, Dmitry, Aleksandr
- 3. Ivanov, Sidorov, Petrov
- 4. 36, 54, 43

ID	FirstName	LastName	Age
001	Vladimir	Ivanov	34
002	Dmitry	Sidorov	56
003	Aleksandr	Petrov	43

Применение колоночных БД

Такая структура удобна для получения данных из базы для анализа.

Например

Если нужно извлечь сумму среднего чека клиента из реляционной СУБД.



В колоночной СУБД можно сразу забрать информацию из нужной колонки.

Популярные колоночные БД



БД временных рядов

В системе данные хранятся в виде связанных пар времени и значения. То есть временная точка служит ключом для получения данных.

Ориентированы на запись и предназначены для обработки постоянного потока входных данных.

Если нужно активно удалять или модифицировать записанные данные, то такие системы лучше не использовать, потому что они будут выполнять эту работу неэффективно

Time	CPU Temp	System Load	Memory Usage %
2019-10-31T03:48:05+00:00	37	0.85	92
2019-10-31T03:48:10+00:00	42	0.87	90
2019-10-31T03:48:15+00:00	33	0.74	87
2019-10-31T03:48:20+00:00	34	0.72	77
2019-10-31T03:48:25+00:00	40	0.88	81
2019-10-31T03:48:30+00:00	42	0.89	82
2019-10-31T03:48:35+00:00	41	0.88	82

Особенности БД временных рядов

Подходят для хранения метрик систем.

Что нужно учесть:

- временная метка — основной параметр
- неэффективно при изменении и удалении данных, иногда это совсем нельзя делать

Применение БД временных рядов

Такая структура удобна для получения данных из базы для анализа.

Например

Нужно извлечь сумму среднего чека клиента из реляционной СУБД

↓

В колоночной СУБД можно сразу забрать информацию из нужной колонки.

Популярные графовые БД



OPENTSDDB



Введение в базы данных

Классификация БД

Skillbox

образовательная платформа

Классификация по модели данных

Модель данных — это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Эти объекты позволяют моделировать структуру данных, а операторы — поведение данных.

Модели данных:

- иерархическая
- сетевая
- реляционная

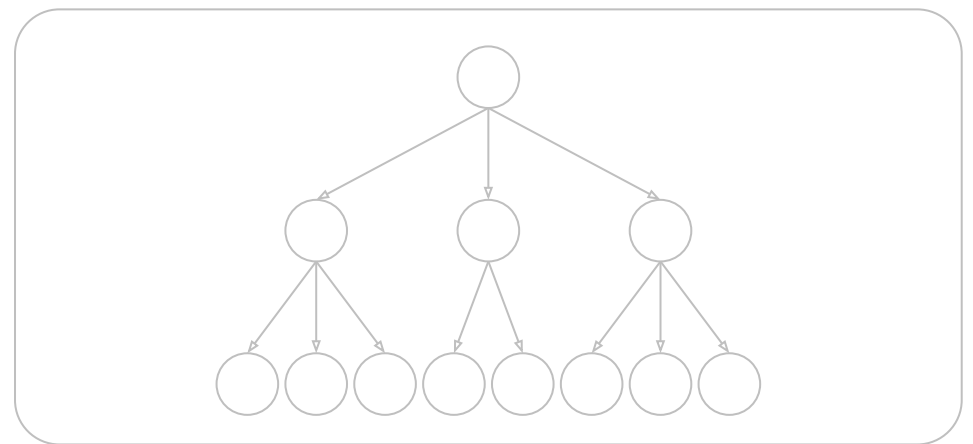
Самая ранняя из них — иерархическая модель.

На её основе появилась сетевая модель, а реляционная появилась самой последней.

Иерархическая модель данных

Информация в иерархической базе организована по принципу древовидной структуры в виде отношений «предок — потомок».

Основные сферы применения: файловые системы и серверы каталогов.



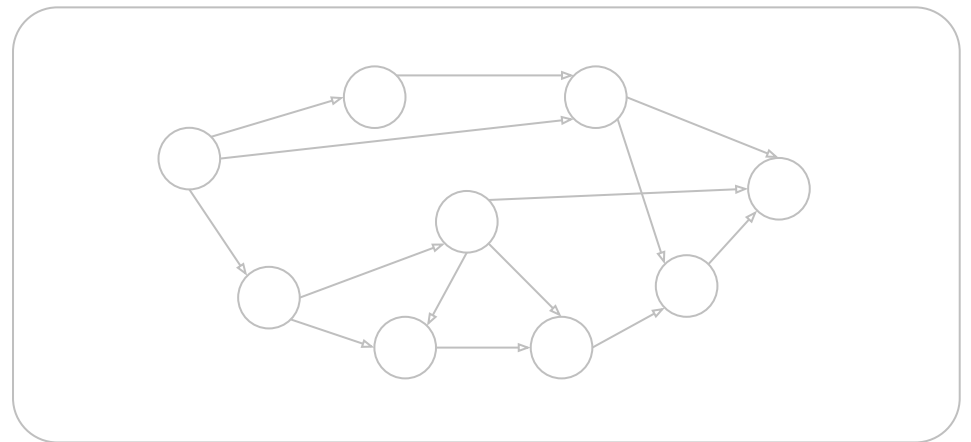
Данные в виде древовидной структуры
в виде отношений «предок — потомок»

- + простота модели для понимания и реализации в виде программы
- множество областей науки и жизни не могут быть описаны только в рамках иерархической модели
- избыточность данных

Сетевая модель данных

Расширяет иерархическую модель:

- данные представлены в виде графа с любым количеством предков и потомков у узла
- допустимо наличие циклов в графе, что запрещено в предыдущей модели



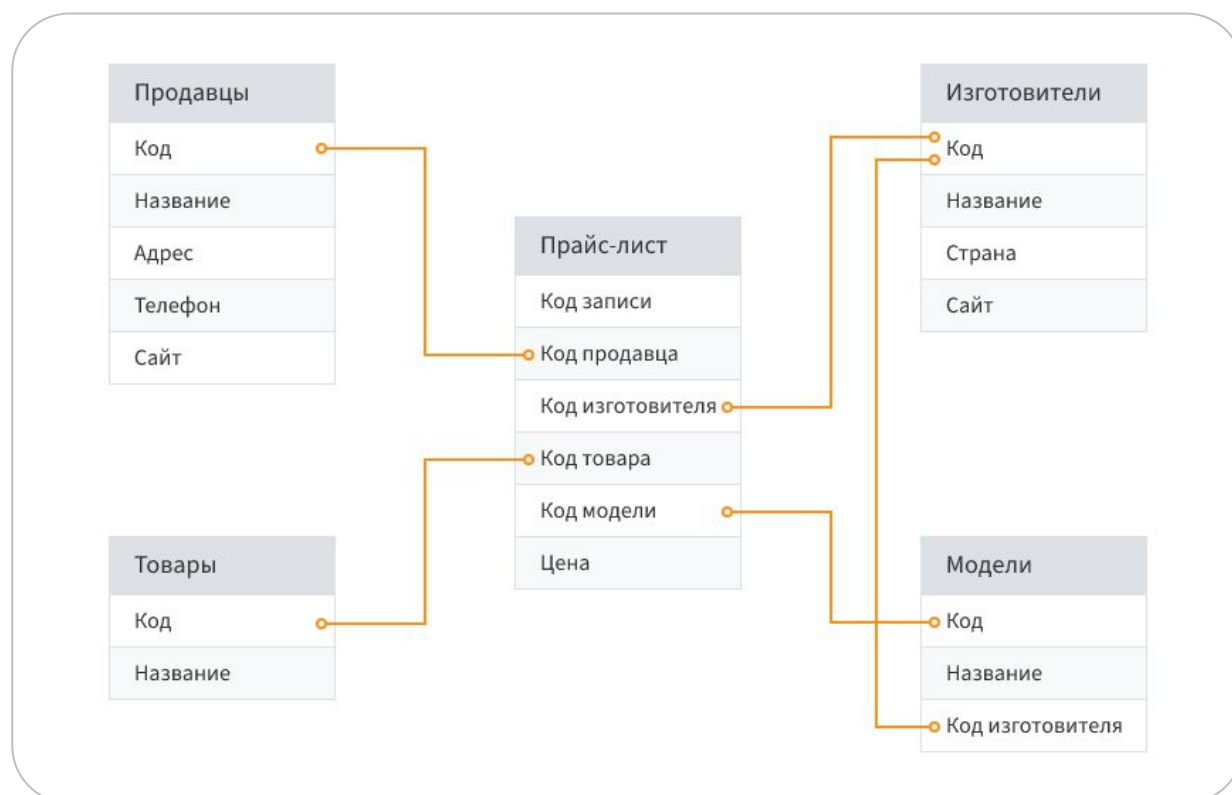
Пример организации данных в сетевой модели. Узлы имеют связь «многие ко многим». Циклы разрешены

Основные области применения: в графических системах и системах пространственной координации объектов.

- + модель может описать намного больше областей с меньшей избыточностью данных
- сложность работы с ней при разработке программ (так как необходимо понимать структуру узлов и рёбер)

Реляционная модель данных

Данные представляют собой набор отношений (таблицы). Каждая таблица — это совокупность строк и столбцов, где строки соответствуют экземпляру объекта, а столбцы — его атрибуты.



Пример организации данных в реляционной модели. Каждый прямоугольник представляет из себя таблицу с атрибутами (колонками) и связи между таблицами

Классификация по доступу пользователей

Различают два типа доступа пользователей: многопользовательский и монопольный доступ. Виды доступа определяются тем, какого рода приложение и какое количество пользователей с ним работает.

Доступ

- **Многопользовательский**

Запрос на обработку данных выполняется клиентом и передаётся по сети на сервер баз данных. Обработка запроса производится на сервере, а после полученные данные передаются обратно клиенту по сети

- **Монопольный**

Для этого типа характерен коллективный доступ к общей базе данных на файловом сервере. При обновлении файла одним из пользователей он блокируется для доступа другим пользователям

Классификация по расположению

База данных может располагаться на собственных серверах компании, на серверах третьих лиц (провайдеров), в облаке или запускаться прямо на мобильном устройстве. Ниже все три случая, в которых они используются.

БД располагаются:

- On-premise (внутри компании)
- Cloud (облако провайдера)
- Mobile device (на мобильном устройстве)

On-premise (локально)

База данных размещается на серверах компании, и компания сама осуществляет её поддержку, администрирование.

On-premise (локально)

Что нужно учесть:

- Возможна индивидуальная настройка БД, а также её доработка под свои нужды
- Данные не передаются 3-им лицам, что увеличивает безопасность данных
- Скорость доступа ещё выше
- Необходимо сопровождение, обновление, администрирование и обеспечение безопасности БД своими силами разработки, что требует экспертизы и вложения средств
- Необходимо иметь собственные сервера, а также заниматься их поддержкой
- При масштабировании системы нужно самим докупать сервера и настраивать их, что более время- и трудозатратно, чем покупка дополнительных вычислительных мощностей у провайдера

Cloud (в облаке)

База данных размещается на серверах облачного провайдера сервисов. Компании предоставляется доступ только к выделенным ресурсам через интернет.

Примеры провайдеров: AWS, Google Cloud Platform, Yandex.Cloud.

Плюсы:

- хорошая масштабируемость
- отсутствие затрат на разработку
- плата только за использованные ресурсы
- скорость внедрения в инфраструктуру



Cloud (в облаке)

База данных размещается на серверах облачного провайдера сервисов. Компании предоставляется доступ только к выделенным ресурсам через интернет.

Примеры провайдеров: AWS, Google Cloud Platform, Yandex.Cloud.

Минусы:

- скорость работы зависит от скорости передачи
- ограничение конфигурируемости
- дополнительные риски по безопасности (перенос данных, непрозрачность инфраструктуры)



На мобильном устройстве (в облаке)

В этом случае база располагается на устройствах пользователя, где запущено приложение. Создаётся она в момент запуска приложения и обычно хранится в одном файле.

Плюсы:

- это легковесное локальное хранилище на устройстве, которое упрощает работу приложения и даёт возможность сохранять данные между сессиями

Минусы:

- данные базы сложно обезопасить, так как база находится на клиенте. Также база может быть удалена, например, при удалении приложения

Примеры — SQLite.

Классификация по распределённости

Базы данных могут быть запущены на одном сервере или быть распределёнными между несколькими серверами.

Классификация по распределённости

Централизованная БД

Располагается на одном сервере или компьютере.

- + Нет проблем с синхронизацией БД на разных машинах, простая реализация транзакций
- + Система проще в запуске, обновлении и администрировании
- Система может масштабироваться только вертикально (увеличение мощности компьютера), что намного хуже, чем горизонтальное масштабирование
- При выходе сервера из строя можно потерять все данные
- Нет возможности роутинга запросов в базу по географическим районам
- Управление безопасностью сосредоточено в одном месте

Классификация по распределённости

Кластерная БД

БД «живёт» на нескольких серверах. На каждом из них сосредоточена только определённая часть данных. Для клиента такой кластер обычно выступает как единое целое.

- + Хорошо масштабируется
- + Хранит большой объём данных
- + Можно делить машины по географическим районам
- Сложность синхронизации
- Необходимость поддержки кластера из серверов

Классификация по типу нагрузки

По типу нагрузки базы данных делятся на OLTP (онлайн-обработка транзакций) и OLAP (онлайн-аналитическая обработка). OLTP собирает, хранит и обрабатывает данные транзакций в режиме реального времени. OLAP использует сложные запросы для анализа агрегированных исторических данных из OLTP-систем.

Классификация по типу нагрузки

OLTP

- Основная нагрузка приходится на **обработку множества небольших транзакций**
- Запросы в транзакциях обычно простые
- Данные не только читаются, но также активно добавляются, обновляются и удаляются
- Строгие требования по скорости (десятки — сотни миллисекунд)
- Размер базы не очень большой

Классификация по типу нагрузки

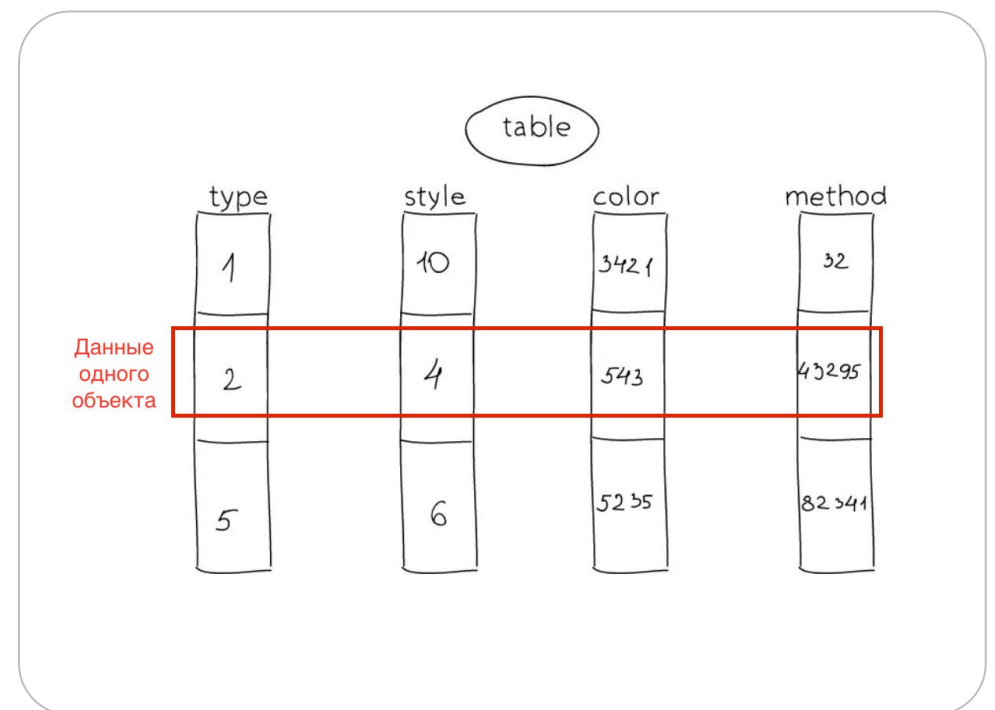
OLAP

- Нагрузка приходится на обработку больших объёмов данных со сложными запросами
- В запросах присутствуют группировки и агрегации данных, подсчёт суммы, среднего и др.
- Время ответа не слишком важно: возврат результата может быть через одну секунду, несколько десятков секунд, а может быть и через несколько минут
- Порядок секунды, десятки секунд (и больше)
- Размер базы очень большой и состоит из множества данных разных OLTP-баз

Классификация по механизму хранения

Колоночное

- Данные хранятся сгруппировано по атрибутам, то есть информация об одном объекте распределена по различным колонкам
- Колоночное хранение данных даёт возможность эффективно производить пакетную обработку атрибутов множества объектов



Классификация по механизму хранения

Строковое

- Данные об объекте хранятся в строке, то есть все его атрибуты записаны в одной строке таблицы
- Быстрый доступ и изменение объекта отлично подходит для ситуаций, когда нужна работа над какой-то группой объектов и нет необходимости в их пакетной обработке
- Но в ситуациях, когда необходимо обновить какую-то часть атрибутов многих объектов, такие системы неэффективны



type	style	color	method
1	10	3421	32
2	4	543	43295
5	6	5235	82341

Введение в базы данных

Окружение для БД

Skillbox

образовательная платформа

Эксплуатационное окружение

БД могут эксплуатироваться в разном окружении

Процесс разработки приложения обычно состоит из нескольких этапов и условно их можно разделить на три типа: продакшн (промышленная эксплуатация), тестовое и дев (разработческое) окружения.

1. Разработка ПО
2. Тестирование
3. Промышленная эксплуатация



Окружение при промышленной эксплуатации

1 Высокий уровень надёжности и защиты

2 Документирование и соответствие
требованиям законодательства

3 Высокая ответственность

Окружение в тестовом контуре

1 Могут быть данные с промышленной эксплуатации

2 Могут быть сниженные требования
по безопасности

3 Более широкий круг пользователей

4 Могут быть утечки промышленных данных
через тестовые площадки

Окружение в среде разработки

- 1 Низкие требования безопасности
- 2 Важна скорость восстановления
- 3 Упрощённые механизмы доступа для разработчика

Эксплуатационное окружение

БД могут эксплуатироваться в разном окружении

- Разные требования и уровни безопасности
- Разные риски и последствия
- Разные пользователи и инфраструктура

