**Ключ : Значение**



У нас есть уникальный ключ который указывает на какое то значение, а сама база – это совокупность этих пар. Эти данные могут быть чем угодно (число, строка, другая пара ключ значение). В таком типе БД нет сложных связей, так как все является просто парой ключ значение

БД такого типа хранят данные в RAM памяти, что обеспечивает быструю скорость работы.

Подходит для кэша или хранения пользовательских сессий.

Например, в catapulto используется кэш Redis, что бы отдать юзеру ценник на перевозку груза (если такой расчет уже был сделан ранее другим пользователем). Соответственно, что бы не тратить производительность системы на передачу запрос другому микросервису и отправку запросов на API курьерских служб, мы делаем запрос в кэш

**Примеры**: Redis, Memcached, DynamoDB

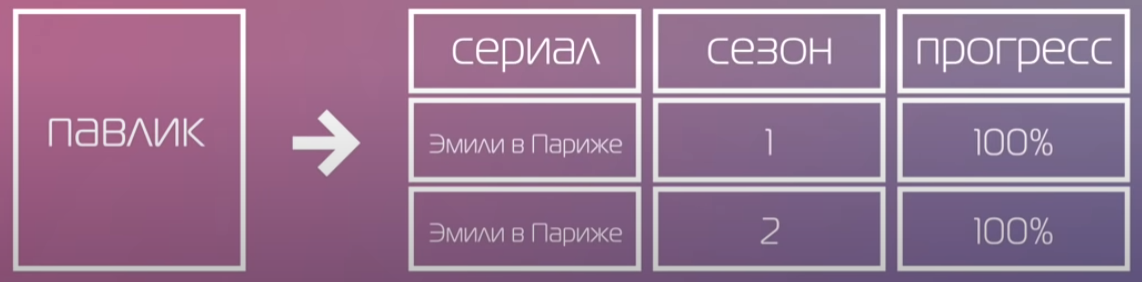
**БД с широкими столбцами – колоночные БД (wide-column)**



То же самое что и БД ключ : значение, только теперь у нас много значений. Данные хранятся на жестких дисках

Используется для хранения логов, аналитики, данных с умных холодильников, чайников и тд

Netflix хранит в себе так историю просмотров юзера



**Примеры**: Casandra, Apache HBASE, ClickHouse

**Документно-ориентированные БД (document DB)**

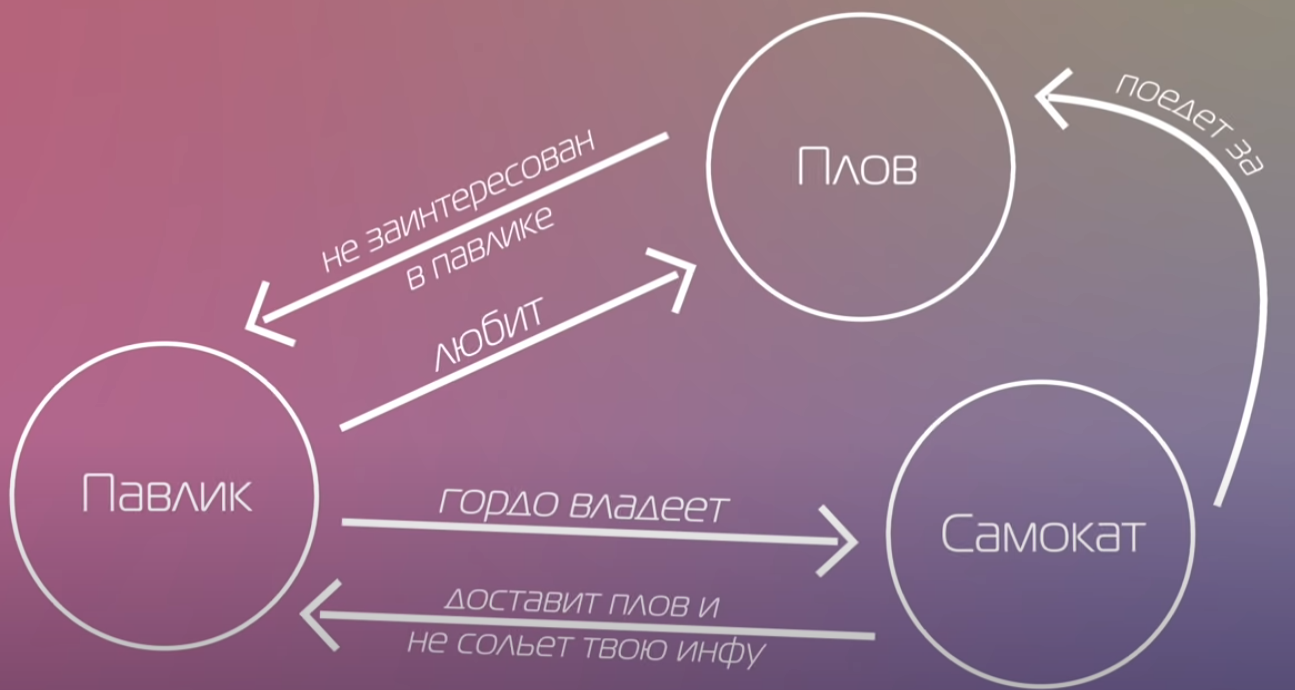
Документы – это набор нескольких пар ключ : значение. Так как это не SQL, то они не требуют схемы, а это значит что мы можем легко добавлять или удалять поля в документе. Документы могут быть вложены и содержать другие документы. Данные хранятся в XML, YAML, JSON форматах. Такая форма хранения идеально подходит к объектам которые используются в приложениях. Мы буквально сразу получаем нужный объект, а в SQL нам пришлось бы делать различные JOIN операции что бы собрать все в единый вид.

Документы можно собирать в коллекции, и их группировать в логическую иерархию, получая что то по типу реляционной БД.

Используется там где есть много контента – Яндекс Дзен, Яндекс Маркет, мобильные приложения, игры, …

**Примеры**: MongoDB, Amazon DynamoDB, CouchDB

**Графовые БД (Graph DB)**



Здесь большее значение уделяется тому как данные связаны между собой. Есть узлы, которые представляют данные и ребра – описывают связь между узлами

Используются в алгоритмах рекомендации, соц сетях и тд

**Примеры**: neo4j, Dgraph

**Поисковые БД (Search-Engine DB)**



Нужны для поиска данных из большого кол-ва источников. Мы создаем документ с текстом внутри, а БД проанализирует весь текст в документах и создаст индексы для этого текста. По сути это работает как указатели, которые мы видим в конце книги, где указывается какой то термин и страница на которой он встречается. Когда пользователь выполняет поиск, то сканируются только индексы, а не все документы в базе

Используются для хранения и анализа логов, полнотекстового поиска

**Примеры:** ElasticSearch, Solr, algolia

**БД временных рядов (Time-Series DB)**



Такие БД оптимизированы для данных с отметками времени

Используются для мониторинга систем, где мы храним значения времени и данные в этот момент

Например загрузка сервера, кол-во подключений

**Примеры**: InfluxDB, Prometheus

Не обязательно использовать разные база данных для каждого случая, так как например тот же Redis может быть:

* Key-value
* Document DB
* Graph DB
* Time series DB