



Преподаватель:

Коляда

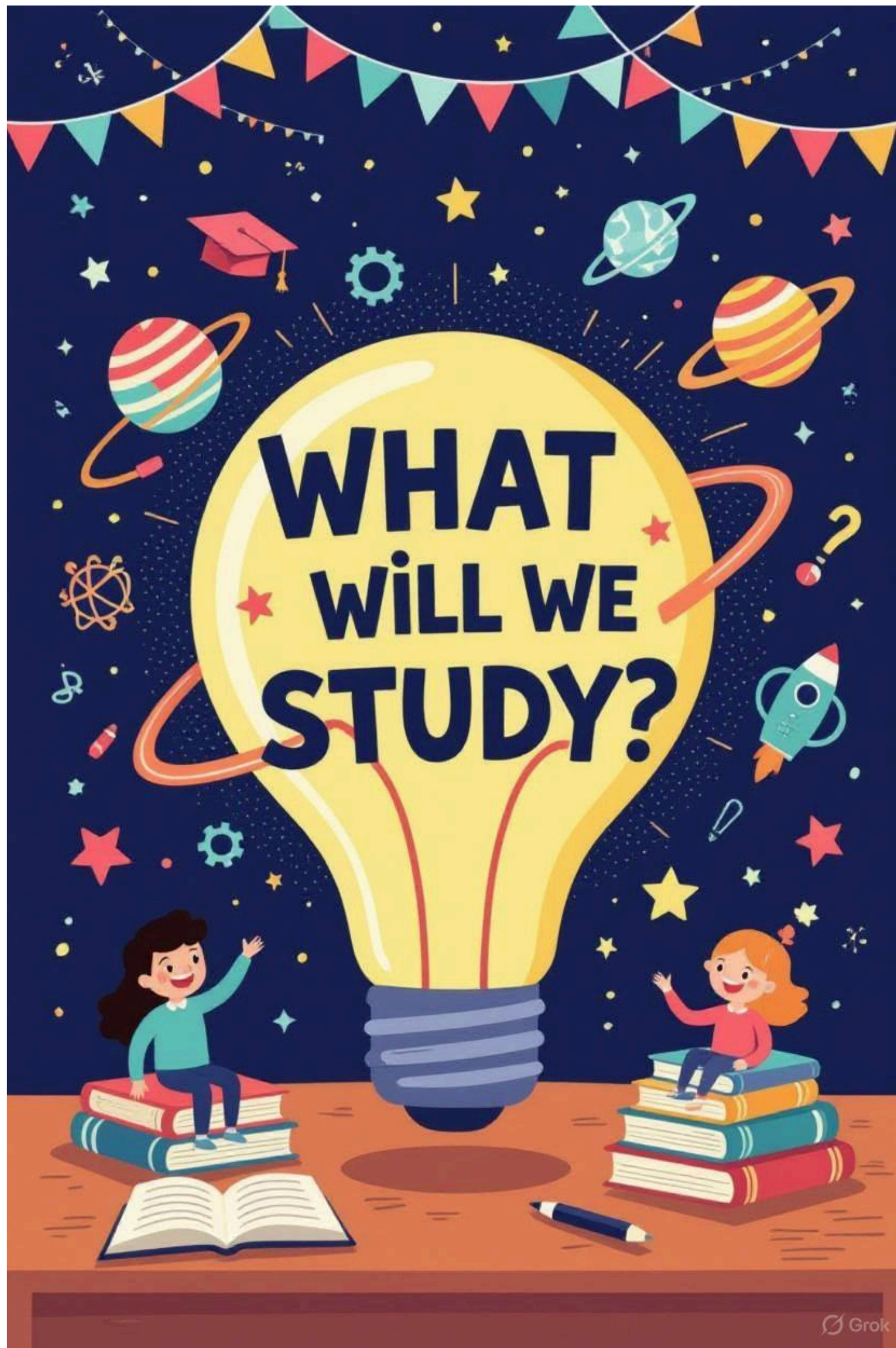
Никита Владимирович

Обратная связь:

- сообщения на inStudy

# ВВЕДЕНИЕ В БАЗЫ ДАННЫХ И СУБД

[КАК ПРИЛОЖЕНИЯ ХРАНЯТ И ОБРАБАТЫВАЮТ ДАННЫЕ]



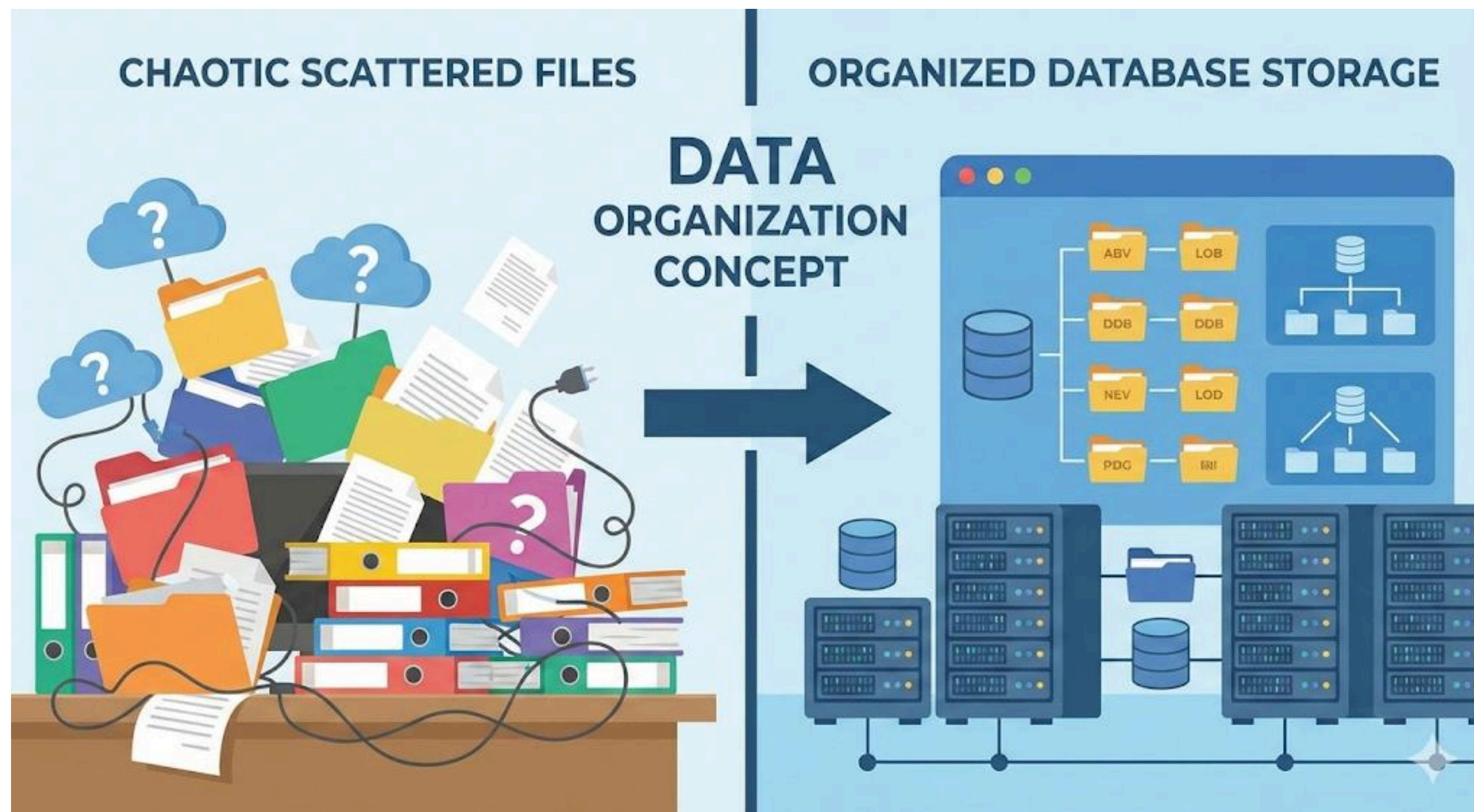
## КЛЮЧЕВЫЕ ТЕМЫ

1. Зачем нужны базы данных — проблема хранения данных
2. Понятие базы данных (что такое БД)
3. Что такое СУБД и примеры СУБД
4. Роль баз данных в современных приложениях
5. Как работает веб-приложение: клиент — сервер — база данных
6. Типы баз данных: реляционные и нереляционные
7. Реляционные базы данных: таблицы и связи
8. Нереляционные базы данных (NoSQL):  
документы, графы, key-value
9. Таблица как основная структура хранения данных
10. Запись и поле: строки и столбцы таблицы
11. Ключи в базе данных: первичный ключ и его роль



# ЗАЧЕМ НУЖНЫ БАЗЫ ДАННЫХ

## ПРОБЛЕМА ХРАНЕНИЯ ДАННЫХ



1. Если хранить данные просто в файлах:
  - сложно искать нужную информацию
  - легко потерять данные
  - нет удобного обновления
  - проблемы с одновременным доступом

2. База данных решает эти проблемы:
  - хранит данные структурировано
  - позволяет быстро находить нужные записи
  - поддерживает одновременную работу многих пользователей
  - обеспечивает безопасность

# ЧТО ТАКОЕ БАЗА ДАННЫХ (БД)

## ПОНЯТИЕ БАЗЫ ДАННЫХ

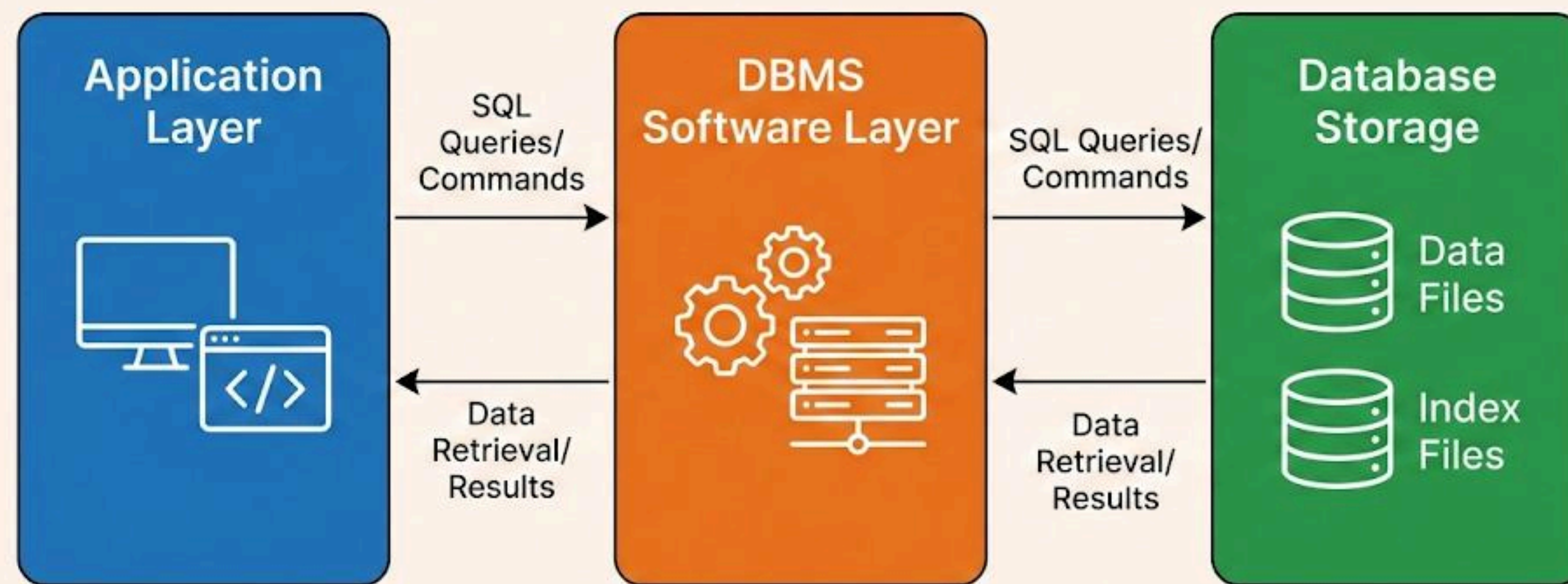


1. База данных – это организованное хранилище информации, предназначенное для долгосрочного хранения и обработки данных.
2. В базе данных могут храниться:
  - пользователи
  - товары
  - заказы
  - комментарии
  - любые другие данные приложения
3. Главная цель БД – быстро и надёжно работать с большими объёмами информации.



# ЧТО ТАКОЕ СУБД

СУБД – СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ



1. СУБД – это программа, которая управляет базой данных.
2. Она отвечает за:
  - добавление и удаление данных
  - поиск и фильтрацию
  - защиту от несанкционированного доступа
  - одновременную работу множества пользователей
3. Примеры СУБД:  
MySQL, PostgreSQL, SQLite, MongoDB
4. Приложение не работает с БД напрямую – оно всегда обращается через СУБД.

# РОЛЬ БД В СОВРЕМЕННЫХ ПРИЛОЖЕНИЯХ

ГДЕ ИСПОЛЬЗУЮТСЯ БАЗЫ ДАННЫХ

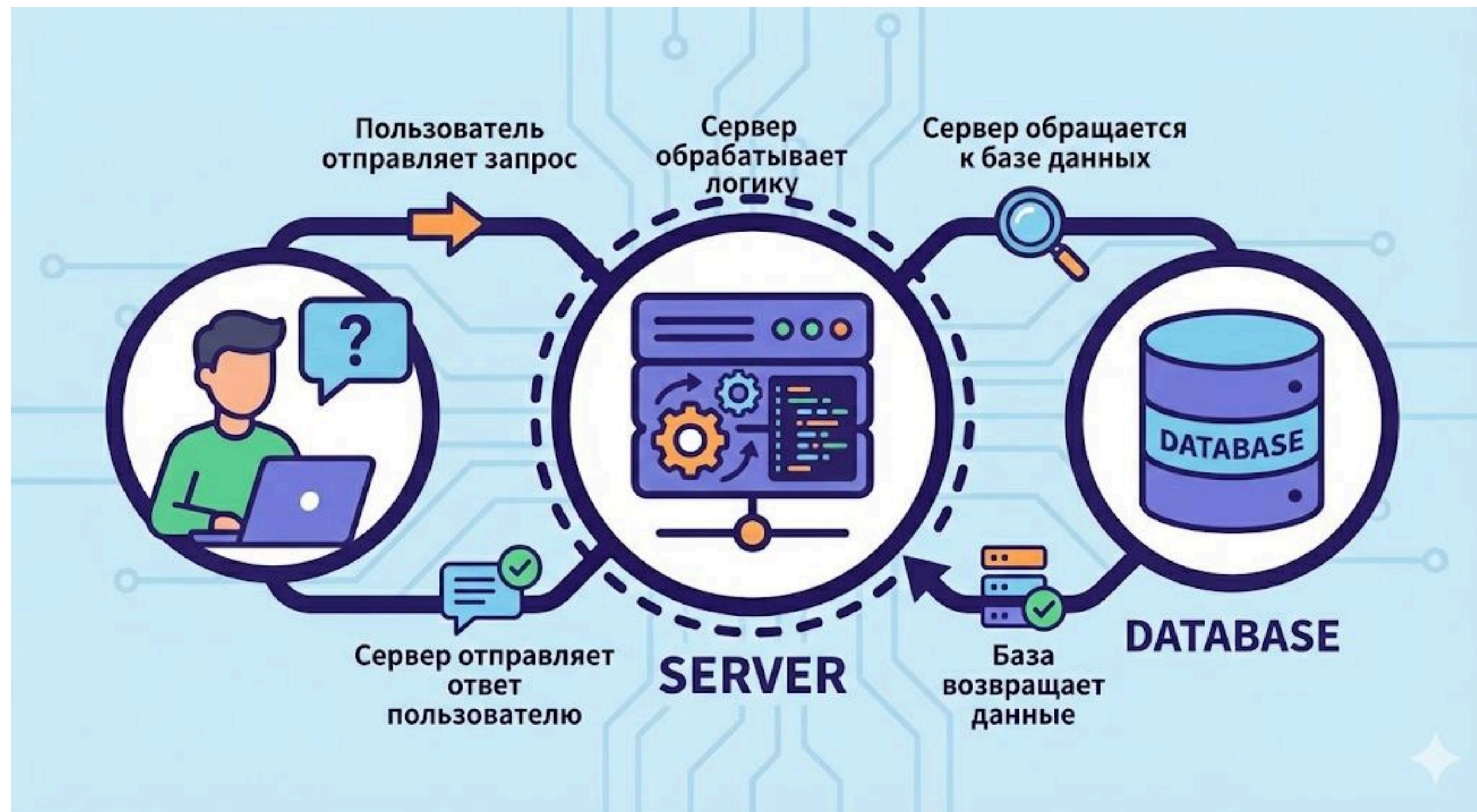


1. Базы данных используются в:
  - интернет-магазинах (товары, заказы, клиенты)
  - социальных сетях (посты, лайки, подписки)
  - банках (счета, транзакции)
  - мобильных приложениях
2. Без базы данных приложение не сможет:
  - запомнить пользователя
  - хранить историю действий
  - работать с большим количеством информации



# КАК РАБОТАЕТ ТИПИЧНОЕ ВЕБ-ПРИЛОЖЕНИЕ

СВЯЗКА: КЛИЕНТ — СЕРВЕР — БАЗА ДАННЫХ



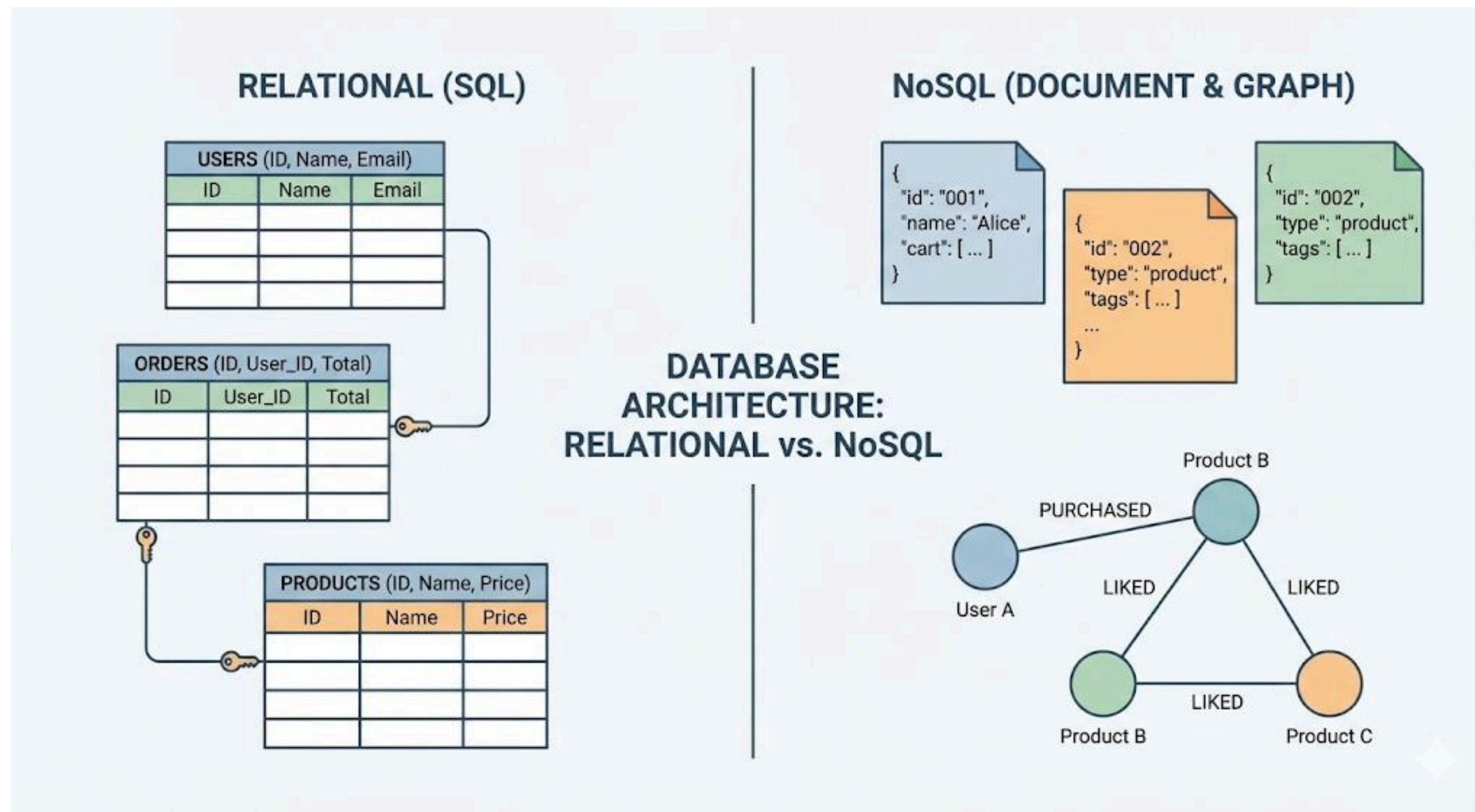
1. Процесс работы выглядит так:

- Пользователь отправляет запрос с браузера или телефона
- Сервер принимает запрос и обрабатывает логику
- Сервер обращается к базе данных
- База возвращает нужные данные
- Сервер отправляет ответ пользователю

2. База данных никогда не доступна напрямую из браузера — только через сервер.

# ТИПЫ БАЗ ДАННЫХ

## ОСНОВНЫЕ ТИПЫ БАЗ ДАННЫХ

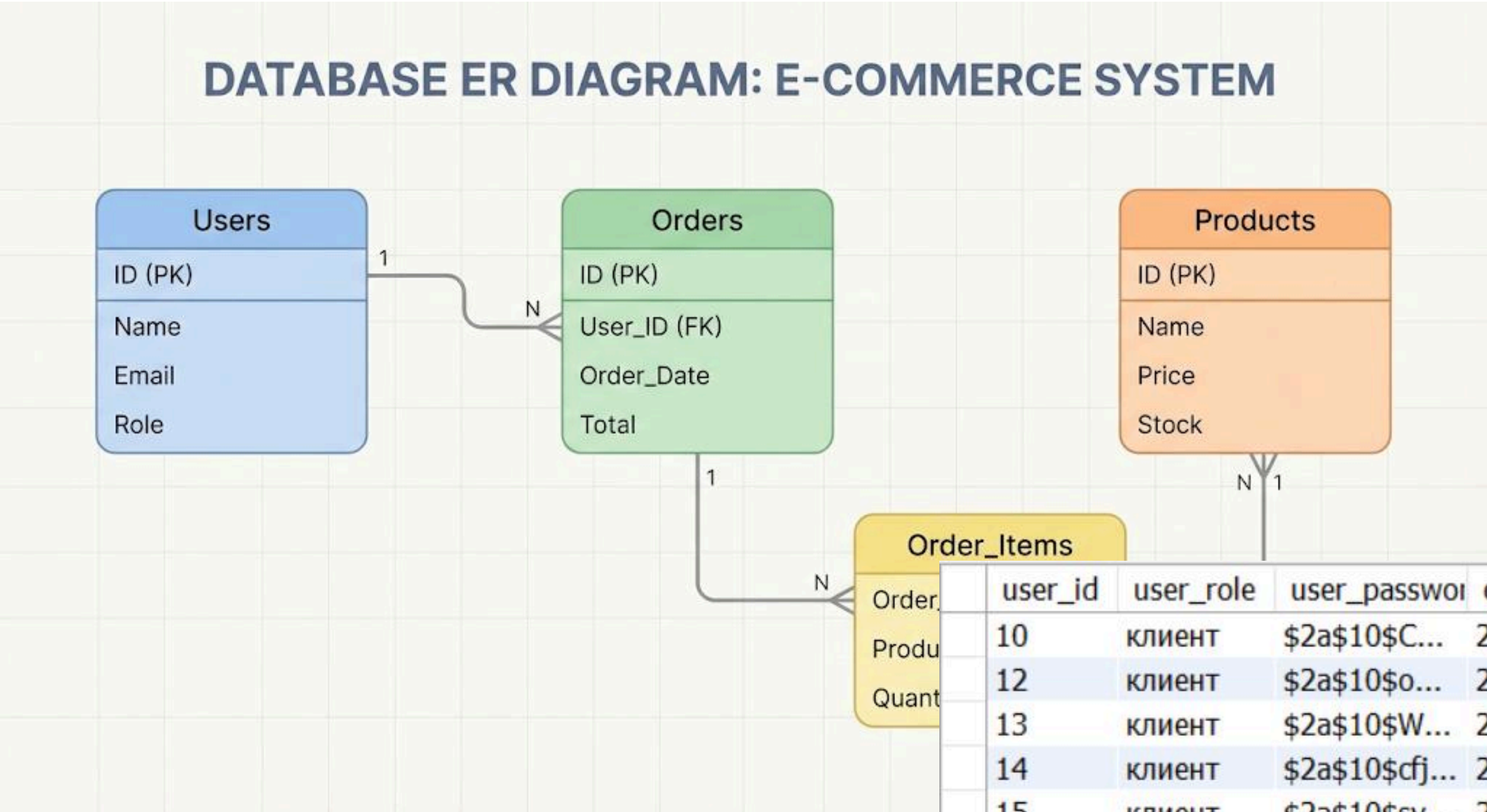


1. Существует два основных подхода к хранению данных:
  - реляционные базы данных
  - нереляционные базы данных (NoSQL)
2. Они отличаются:
  - структурой данных
  - способом хранения
  - типами запросов
  - областью применения
3. Выбор типа зависит от задач проекта.



# РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

## ТАБЛИЦЫ И СВЯЗИ



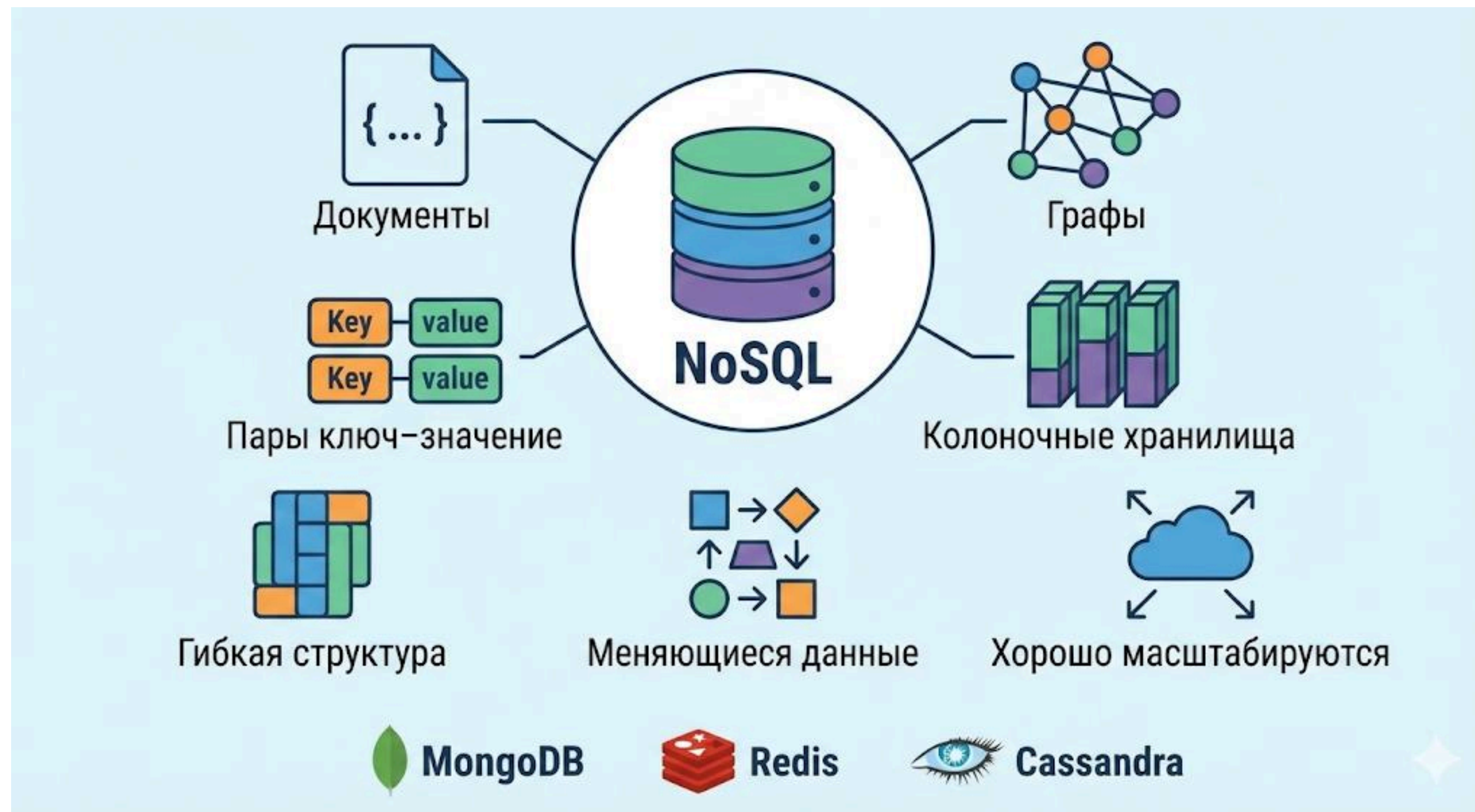
1. В реляционных БД данные хранятся в таблицах.
2. Каждая таблица:
  - состоит из строк и столбцов
  - описывает один тип объектов (например, Users или Orders)
3. Таблицы могут быть связаны между собой:
  - пользователь → заказы
  - заказ → товары
4. Примеры: MySQL, PostgreSQL, Oracle  
Основной язык запросов – SQL.

	user_id	user_role	user_password	created_at	user_name	user_email
	10	клиент	\$2a\$10\$C...	2025-02-08 14:45:58	test	login
	12	клиент	\$2a\$10\$o...	2025-02-08 15:38:22	test	email@test.ru
	13	клиент	\$2a\$10\$W...	2025-02-12 13:46:22	test	email@2test.ru
	14	клиент	\$2a\$10\$cfj...	2025-02-12 13:48:08	test	email@3test.ru
	15	клиент	\$2a\$10\$sv...	2025-02-12 13:48:54	test	email@4test.ru
	16	клиент	\$2a\$10\$6e...	2025-02-24 10:10:37	Ivan	email@email.com
	18	клиент	\$2a\$10\$b...	2025-02-24 10:49:08	Ivan2	email@email.com2
	19	клиент	\$2a\$10\$C...	2025-02-24 11:17:33	Ivan3	email@email.com3
▶	21	клиент	\$2a\$10\$Ij...	2026-01-06 17:45:44	test	3test@email.com
*	NULL	NULL	NULL	NULL	NULL	NULL



# НЕРЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ (NOSQL)

ГИБКИЕ СТРУКТУРЫ ХРАНЕНИЯ



1. В NoSQL данные не обязательно хранятся в таблицах.

2. Возможные форматы:

- документы (похожие на JSON)
- пары ключ-значение
- графы
- колоночные хранилища

3. Преимущества:

- гибкая структура
- удобно работать с меняющимися данными
- хорошо масштабируются

4. Примеры: MongoDB, Redis, Cassandra.



# ТАБЛИЦА КАК СТРУКТУРА ДАННЫХ

ОСНОВНАЯ СТРУКТУРА В РЕЛЯЦИОННЫХ БД

DATABASE TABLE STRUCTURE: ROWS & COLUMNS

ID (PK)	First Name	Last Name	Role
101	Jane	Doe	Admin
102	John	Smith	User
103	Emily	White	Editor

- 1. Таблица – это набор данных, организованный в виде сетки.
- 2. Состоит из:
  - столбцов (поля)
  - строк (записи)
- 3. Каждая строка – отдельный пользователь.

user_id	user_role	user_passwoi	created_at	user_name	user_email
10	клиент	\$2a\$10\$C...	2025-02-08 14:45:58	test	login
12	клиент	\$2a\$10\$o...	2025-02-08 15:38:22	test	email@test.ru
13	клиент	\$2a\$10\$W...	2025-02-12 13:46:22	test	email@2test.ru
14	клиент	\$2a\$10\$cfj...	2025-02-12 13:48:08	test	email@3test.ru
15	клиент	\$2a\$10\$sv...	2025-02-12 13:48:54	test	email@4test.ru
16	клиент	\$2a\$10\$6e...	2025-02-24 10:10:37	Ivan	email@email.com
18	клиент	\$2a\$10\$b...	2025-02-24 10:49:08	Ivan2	email@email.com2
19	клиент	\$2a\$10\$C...	2025-02-24 11:17:33	Ivan3	email@email.com3
21	клиент	\$2a\$10\$Ij...	2026-01-06 17:45:44	test	3test@email.com
HULL	HULL	HULL	HULL	HULL	HULL
*					

# ЗАПИСЬ И ПОЛЕ

ИЗ ЧЕГО СОСТОИТ ТАБЛИЦА

ID	Name	Role	Team
001	John	Admin	Green
002	Bob	Editor	Red
003	Kbrkey	Editor	Green
004	Pdnion	Viewer	Green

← Record (Row)

↑ Field (Column)

1. Запись – это одна строка таблицы.

Она описывает один объект.

2. Поле – это один столбец таблицы.

Оно хранит одно свойство объекта.

3. Например:

Запись – конкретный пользователь

Поля – имя, email, пароль, дата  
регистрации

	user_id	user_role	user_passwoi	created_at	user_name	user_email
	10	клиент	\$2a\$10\$C...	2025-02-08 14:45:58	test	login
	12	клиент	\$2a\$10\$o...	2025-02-08 15:38:22	test	email@test.ru
	13	клиент	\$2a\$10\$W...	2025-02-12 13:46:22	test	email@2test.ru
	14	клиент	\$2a\$10\$cfj...	2025-02-12 13:48:08	test	email@3test.ru
	15	клиент	\$2a\$10\$sv...	2025-02-12 13:48:54	test	email@4test.ru
	16	клиент	\$2a\$10\$6e...	2025-02-24 10:10:37	Ivan	email@email.com
	18	клиент	\$2a\$10\$b...	2025-02-24 10:49:08	Ivan2	email@email.com2
	19	клиент	\$2a\$10\$C...	2025-02-24 11:17:33	Ivan3	email@email.com3
▶	21	клиент	\$2a\$10\$Ij...	2026-01-06 17:45:44	test	3test@email.com
*	NULL	NULL	NULL	NULL	NULL	NULL



# КЛЮЧИ В БАЗЕ ДАННЫХ

## ЗАЧЕМ НУЖЕН ПЕРВИЧНЫЙ КЛЮЧ

### PRIMARY KEY (ID)

Unique identifier for each record.



ID	First Name	Last Name	Email
001			
002			
003			
004			
005			

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
user_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
user_role	ENUM('клиент', 'ss', '...')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'клиент'
user_password	VARCHAR(345)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
created_at	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP
user_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
user_email	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Ключ – это поле, которое однозначно определяет запись.
2. Primary Key (первичный ключ):
  - уникален
  - не может повторяться
  - часто используется числовой id
3. Ключи нужны для:
  - быстрого поиска
  - связи таблиц между собой
  - обеспечения целостности данных



*ВОПРОСЫ?*