

Системи за управление на бази от данни (СУБД) □



1. Видове бази данни
 - Релационни БД
 - Нерелационни БД
2. Типове данни
3. Какво е СУБД?



Видове бази данни

SQL vs. NoSQL

- Работим с **два вида** бази данни:

- **Реляционни БД**

- Съдържат данни в **таблицы** + **релации** (връзки)
 - Използват езика **SQL** за заявки



- **NoSQL (нереляционни) БД**

- Имат **колекции** от документи или двойки ключ-стойност



Примери за SQL и NoSQL бази данни

■ SQL бази данни:

- MySQL
- PostgreSQL
- Oracle
- Microsoft SQL
- SQLite and Web SQL

■ NoSQL бази данни:

- MongoDB
- Redis
- Google BigTable
- Amazon DynamoDB
- Azure Cosmos DB
- Cassandra



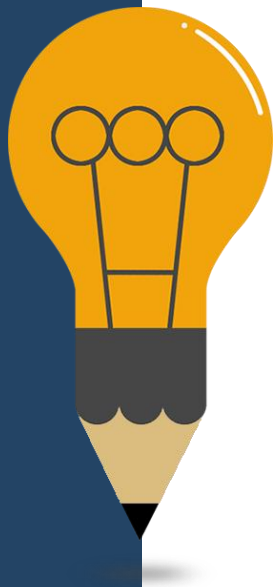


Реляционни бази данни

Таблици, релации и SQL

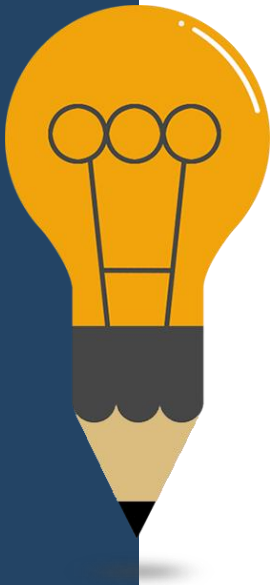
Релационни бази данни (1)

- Релационните (**SQL**) бази данни организират данните в **таблицы**
- Таблиците имат строга структура (**колони** със зададени **типове на данни**)
- Могат да имат **връзки** към други таблици

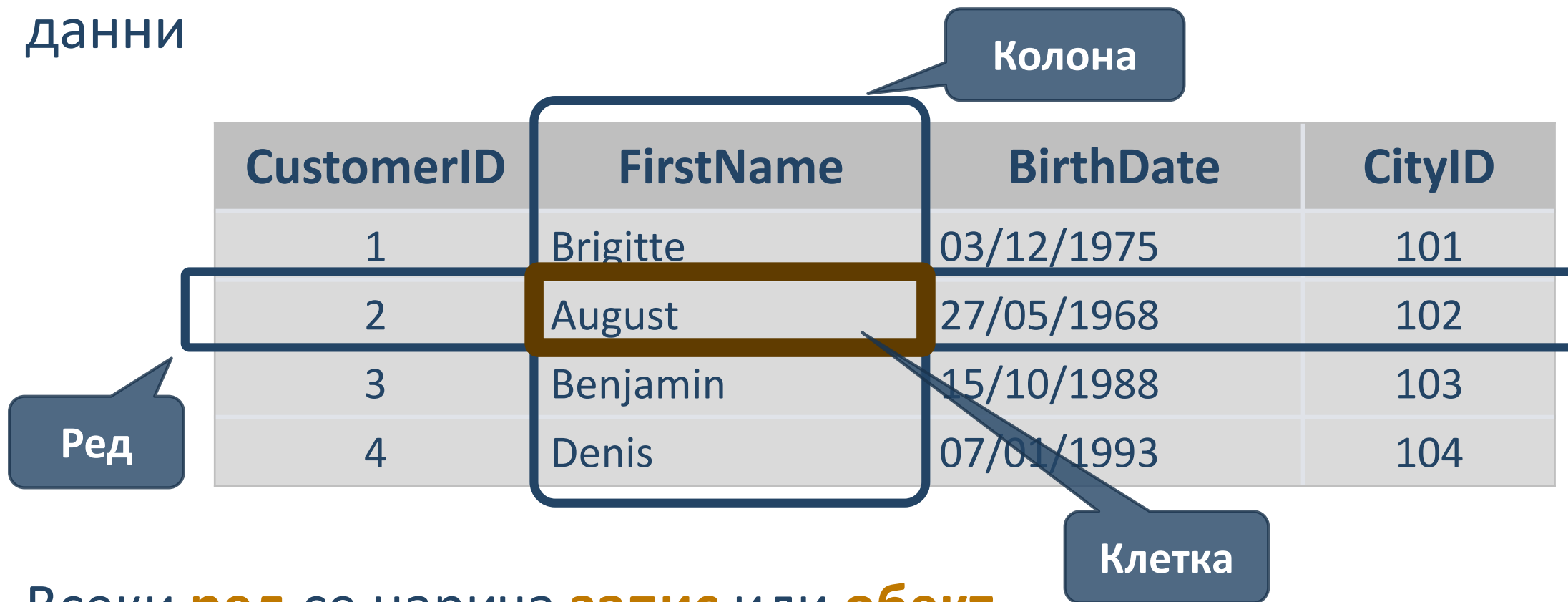


Релационни бази данни (2)

- **Релационните** бази данни използват езика **SQL** за дефиниране и манипулиране на данни.
- Изключително мощен за сложни заявки
- **Релационните бази данни** са най-използваната технология за управление на данни.



- **Таблиците** са основният **градивен елемент** на релационните бази данни



CustomerID	FirstName	BirthDate	CityID
1	Brigitte	03/12/1975	101
2	August	27/05/1968	102
3	Benjamin	15/10/1988	103
4	Denis	07/01/1993	104

- Всеки **ред** се нарича **запис** или **обект**
- Колоните (**полета**) определят **вида** на данните

- Релационните данни се съхраняват в една или повече **таблици**, които може да имат:
 - **Уникален ключ**, идентифициращ всеки ред
 - **Външни ключове**, определящи **връзки**



Релационният модел на БД – пример

Предмети

ID	Order ID	Name	Quantity	Price
5	1	Table	1	200.00
6	1	Chair	1	123.12

Клиенти

ID	Name	Email
5	Peter	peter@gmail.com
6	Jayne	jayne@gmail.com



ID	Customer ID	Date	Total Price
1	5	11/1/17	323.12
2	1	11/15/17	13.99

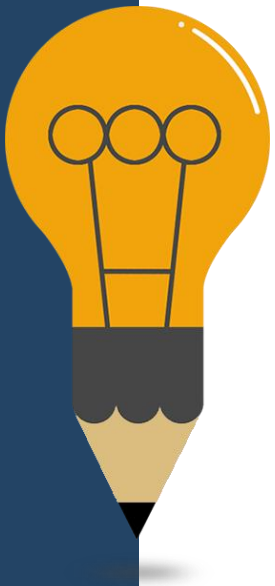


Нерелационни бази данни

NoSQL бази данни и JSON документи

Нерелационни (NoSQL) бази данни

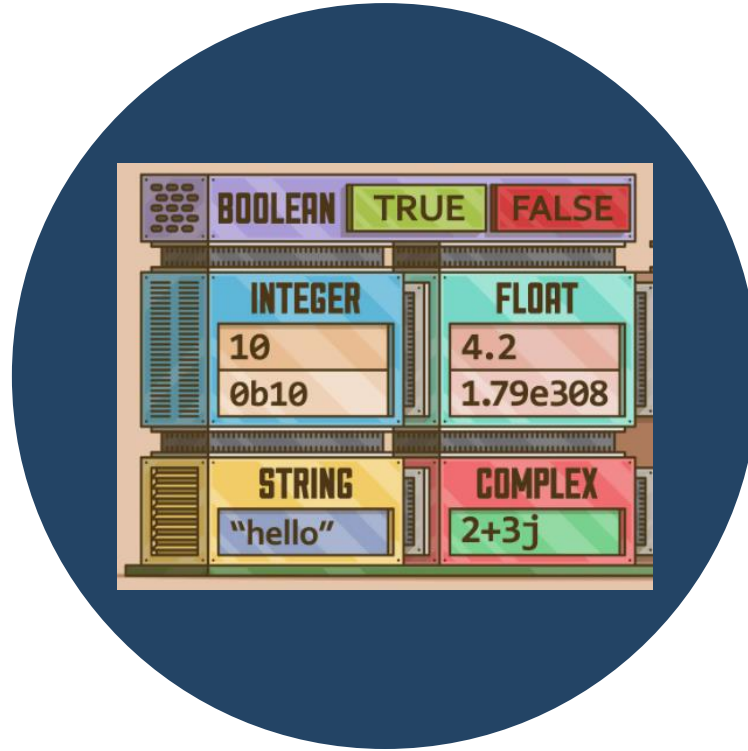
- **Нерелационните** бази данни имат динамична схема за **неструктурирани** данни
- Данните могат да се съхраняват по няколко начина:
 - Документен (JSON store)
 - Чрез колони (table store)
 - Базиран на граф
 - Двойки ключ-стойност



- **NoSQL бази данни** не използват таблици и SQL
 - Вместо това използват **колекции от документи** или **двойки ключ-стойност**
- **Мащабируеми** и с **висока производителност**
- Примери: **MongoDB, Cassandra, Redis**, etc.

```
{  
  "_id": ObjectId("59d3fe7ed81452db0933a871"),  
  "email": "peter@gmail.com",  
  "age": 22  
}
```

Пример за JSON
документ в MongoDB



Типове данни в SQL Server

- Числови
 - Целочислени типове данни:
 - **BIT** (1-bit), **TINYINT** (8-bit), **SMALLINT** (16-bit)
 - **INT** (32-bit), **BIGINT** (64-bit)
 - Типове данни с плаваща запетая:
 - **FLOAT**, **REAL**, **DECIMAL**
- Текстови
 - **CHAR(size)** – низ с фиксиран размер
 - **VARCHAR(size)** – символен низ с променлив размер
 - **NCHAR(size)** – Unicode низ с фиксиран размер
 - **NVARCHAR(size)** – Unicode символен низ с променлив размер
- **NULL** – празна стойност

- Двоични/бинарни данни
 - **BINARY(size)** – поредица от битове с фиксирана дължина
 - **VARBINARY(size)** – поредица от битове, 1-8000 байта или **MAX** (2GB)
- Дата и време (типове данни за дата и час)
 - **DATE** – дата в диапазона от 0001-01-01 до 9999-12-31
 - **DATETIME** – дата и час с точност 1/300 сек
 - **DATETIME2** – тип, който има по-голям период от време
 - **SMALLDATETIME** – дата и час (с точност до 1 минута)
 - **TIME** – определя час от деня (без часова зона)
 - **DATETIMEOFFSET** – дата и час, които имат часова зона

Дата и време в SQL Server

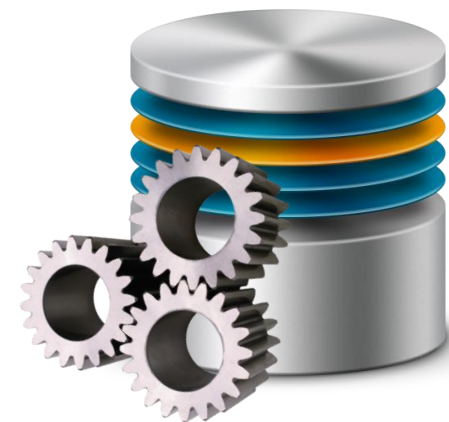
DATA TYPE	① RANGE OF VALUES	② ACCURACY	③ STORAGE SPACE
<i>SMALLDATETIME</i>	01/01/1900 to 06/06/2079	<u>1 minute</u>	4 bytes
<i>DATETIME</i>	01/01/ <u>1753</u> to 12/31/9999	0.00333 seconds	8 bytes
<i>DATETIME2</i>	01/01/ <u>0001</u> to 12/31/9999	100 <u>nanoseconds</u>	<u>6 to 8 bytes</u>
<i>DATETIMEOFFSET</i>	01/01/0001 to 12/31/9999	100 nanoseconds	8 to 10 bytes
<i>DATE</i>	01/01/0001 to 12/31/9999	1 day	<u>3 bytes</u>
<i>TIME</i>	00:00:00.0000000 to 23:59:59.9999999	100 nanoseconds	3 to 5 bytes



Системи за управление на бази данни

СУБД

- **СУБД** е софтуер, използван за **дефиниране**, **манипулиране**, **извличане** и **управление** на данни в база данни
 - СУБД **съхранява и управлява** самите данни, формата им, имената на полетата и типовете данни, структурата на записа и файловата структура
 - **Източник / доставчик** на данни



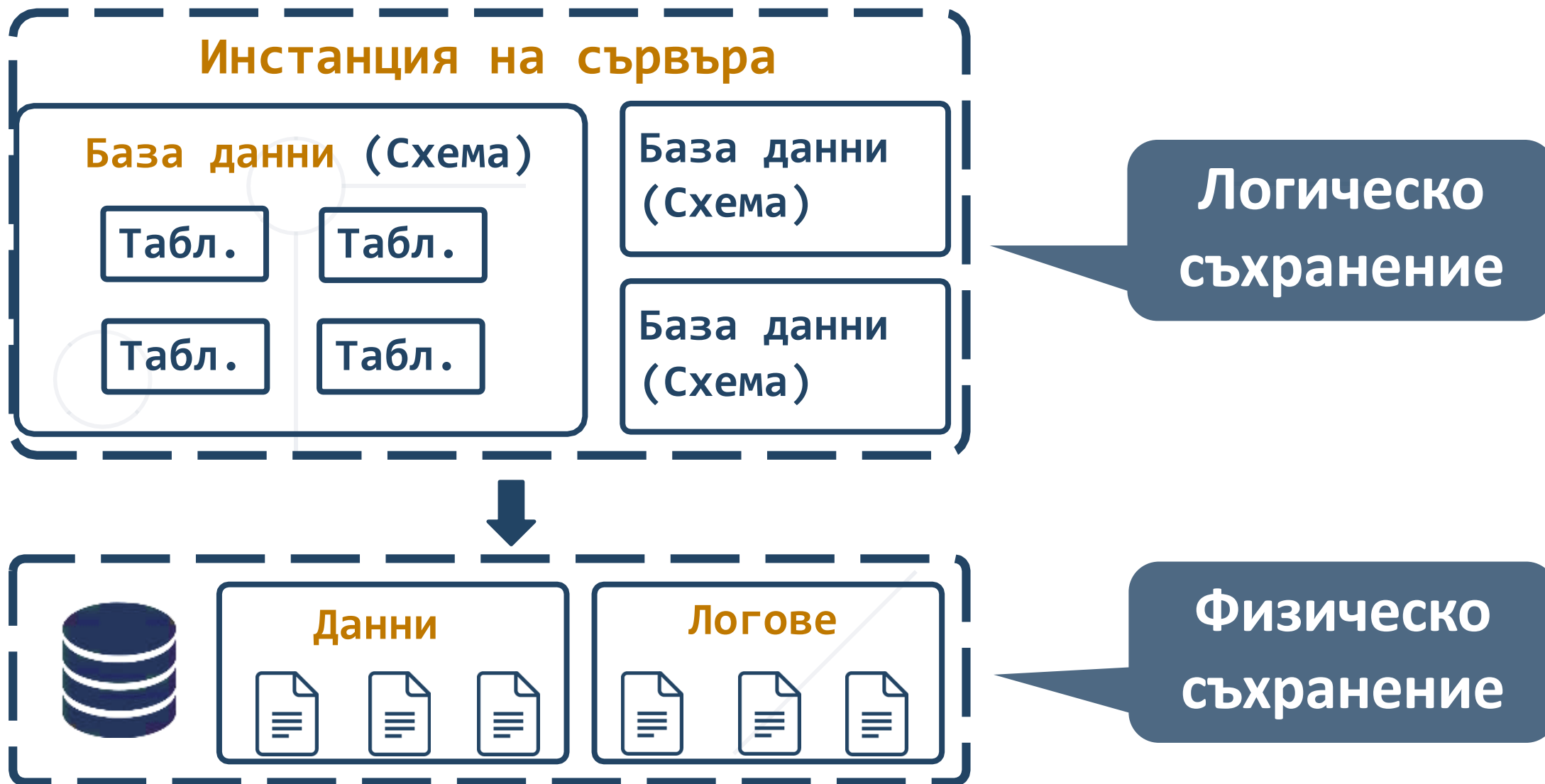
СУБД и поток от данни (схема на база от данни)

- СУБД използват модела **клиент-сървър**:

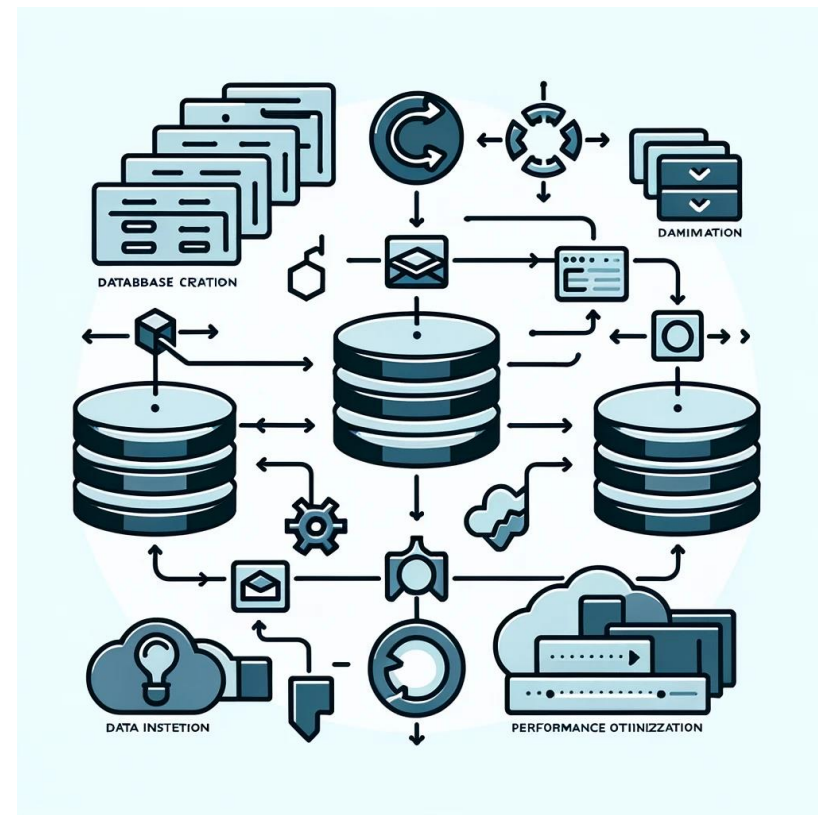


- СУБД примери (сървъри за бази данни):
 - MySQL, MS SQL Server, Oracle, PostgreSQL
 - MongoDB, Cassandra, Redis, HBase
 - Amazon DynamoDB, Azure Cosmos DB





- **Създаване** на базата данни
- **Създаване** на таблици
- **Вмъкване** на данни
- **Модифициране** на данни
- **Извличане** на данни
- **Управление** на **транзакции**
- **Оптимизация** на **производителността**
- **Сигурност** и **резервно копие**



- Управлява **съхранението** и **достъпа** до **файлове**, които съдържат **бази данни** и свързаната с тях **информация**
- Централизиран ресурс, който осигурява:
 - Централизирано съхранение
 - Контрол на достъпа
 - Ефективност и производителност
 - Споделяне на данни
 - Масшабируемост



- Разпределена система == СУБД, която:
 - Не се съхранява на едно място, а е **разпределена** между **различни узли**
 - Узлите се свързват чрез **мрежа**
 - Позволява на базата данни да бъде **разпределена** или **репликирана** между **различни точки** в **мрежата**
 - Подобрява **достъпността**, **надеждността** и **мащабируемостта**

- **Абстракцията** в СУБД е **ключова концепция**, която улеснява взаимодействието между потребителите и самите бази данни
- **Скрива** сложността на физическото съхранение на данните
- **Представя** само **необходимата** информация на различните **нива**:
 - Външно
 - Концептуално
 - Вътрешно

- Начин, по който **потребителят** вижда данните
- Това е **най-високото ниво** от трите нива на **абстракция**
- Външните нива на абстракция могат да представляват различни "**изгледи**" (**views**) на базата данни
 - Например таблица с данни, която е специфична за нуждите на отдел в организацията

- Посредник между **външните** и **вътрешните** нива на абстракция
- Описва **структурата** на цялата **база данни** за **всички** потребители
- Определя какви **данни** се **съхраняват** в базата данни и **връзките** между тях, но без да влиза в детайли как точно тези данни се съхраняват физически

- **Най-ниското** ниво и описва **физическото** съхранение на данните в базата
- Включва **физическите пътища** за достъп до данните, както и **структурите за съхранение** като **индекси** и **хеш-таблиците**
- Занимава се с **оптимизирането** на **производителността** на базата данни и ефективното използване на хардуерните ресурси

Какво е клиентско приложение на бази от данни?

- **Софтуер**, който позволява на крайните потребители да **взаимодействат** с база данни
- Такова приложение служи като **посредник** между **потребителя** и **системата** за **управление** на **бази данни**
 - Предоставя **интерфейс** за извършване на **различни операции** над **данните**:
 - **Търсене**
 - **Създаване**
 - **Промяна**
 - **Изтриване**



- **Релационни бази данни:** таблици и релации със строга структура
- **Нерелационни бази данни:** колекции от документи
- **Типове данни:**
 - Числови
 - Бинарни
 - Текстови
 - Дати
- **СУБД:** съхранява и управлява данните



Въпроси?