# Лекция: Введение в базы данных (БД)

## 1. Что такое система управления базами данных (СУБД)?

СУБД (Система управления базами данных) — это программное обеспечение, которое используется для создания, управления, обработки и анализа данных в базах данных. СУБД упрощает взаимодействие пользователей и приложений с данными, предоставляя интерфейс для их обработки.  
  
Основные функции СУБД:  
1. Создание базы данных: Определение структуры и создание таблиц, индексов, представлений и других объектов.  
2. Хранение данных: Эффективное хранение данных в памяти и на диске.  
3. Обеспечение безопасности: Ограничение доступа к данным через учетные записи, пароли и роли.  
4. Многопользовательский доступ: Возможность работы нескольких пользователей одновременно.  
5. Обеспечение целостности данных: Поддержка ограничения уникальности, связей между таблицами и других правил.  
6. Резервное копирование и восстановление: Предотвращение потери данных.  
  
Примеры СУБД:  
- Реляционные: MySQL, PostgreSQL, SQLite, Oracle, Microsoft SQL Server.  
- Документо-ориентированные: MongoDB, CouchDB.  
- Графовые: Neo4j.  
- Ключ-значение: Redis, DynamoDB.

## 2. Основные понятия СУБД

2.1. База данных  
База данных — это организованная совокупность данных, которые хранятся и управляются СУБД. Данные структурированы таким образом, чтобы их можно было легко находить и обрабатывать.  
  
2.2. Таблица  
Таблица — это базовая структура реляционной базы данных. Она состоит из строк и столбцов:  
- Строки (записи): Хранят конкретные данные.  
- Столбцы (поля): Хранят определённые атрибуты.

Пример таблицы 'Ученики':  
  
| ID | Имя | Фамилия | Возраст | Класс |  
|----|------------|---------------|-------------|-------|  
| 1 | Иван | Иванов | 10 | 4А |  
| 2 | Мария | Смирнова | 9 | 3Б |  
  
2.3. Ключи  
- Первичный ключ (Primary Key): Поле, которое уникально идентифицирует запись в таблице (например, ID).  
- Внешний ключ (Foreign Key): Поле, которое указывает на первичный ключ в другой таблице для создания связей между таблицами.  
  
2.4. SQL  
SQL (Structured Query Language) — язык для работы с реляционными базами данных. Он используется для:  
- Создания и изменения объектов (таблиц, баз данных и т. д.).  
- Обработки данных (добавление, удаление, обновление).  
- Запросов данных (выборка информации по критериям).  
  
Пример запроса:  
SELECT Имя, Фамилия FROM Ученики WHERE Класс = '4А';

## 3. Создание баз данных, таблиц и нормализация

3.1. Создание базы данных  
База данных создаётся с использованием команды SQL:  
  
CREATE DATABASE Имя\_БД;  
  
Пример:  
CREATE DATABASE Школа;  
  
3.2. Создание таблиц  
Таблицы создаются для хранения данных. Пример создания таблицы 'Ученики':  
  
CREATE TABLE Ученики (  
 ID INT PRIMARY KEY,  
 Имя VARCHAR(50) NOT NULL,  
 Фамилия VARCHAR(50),  
 Возраст INT,  
 Класс VARCHAR(10)

3.3. Нормализация  
Нормализация — это процесс проектирования структуры базы данных для устранения избыточности и обеспечения целостности данных.  
  
Основные формы нормализации:  
1. Первая нормальная форма (1NF): Удаление повторяющихся данных, каждая ячейка таблицы должна содержать одно значение.  
2. Вторая нормальная форма (2NF): Удаление зависимостей между неключевыми полями и составным ключом.  
3. Третья нормальная форма (3NF): Устранение транзитивных зависимостей между полями.

## 4. Виды баз данных

4.1. Реляционные базы данных  
- Хранят данные в виде таблиц.  
- Связи между таблицами устанавливаются через ключи.  
- Используют SQL.  
Пример: MySQL, PostgreSQL.  
  
4.2. Документо-ориентированные базы данных  
- Хранят данные в виде документов (JSON, BSON).  
- Подходят для работы с неструктурированными данными.  
Пример: MongoDB, CouchDB.  
  
4.3. Ключ-значение базы данных  
- Хранят данные в парах "ключ-значение".  
- Применяются для хранения кэша или работы с большими объемами данных.  
Пример: Redis, DynamoDB.  
  
4.4. Графовые базы данных  
- Хранят данные в виде узлов (объектов) и связей между ними.  
- Подходят для анализа социальных сетей, работы с рекомендациями.  
Пример: Neo4j.

## Заключение

СУБД упрощают управление данными, обеспечивая безопасность, целостность и доступность. Нормализация структуры данных помогает устранить избыточность и улучшить производительность. Выбор типа базы данных зависит от задач: реляционные подходят для структурированных данных, а документо-ориентированные и графовые для сложных взаимосвязей.