

# Базы данных

## ЛЕКЦИЯ 1

# **Введение в базы данных. СУБД**

Алматы, 2023

# Информация о курсе

- CSE6792 Базы данных, 5 (1/1/1) кредитов
- Лекции - 15 часов (1ч/нед)
- Лабораторные работы (база данных университета) - 30 часов (2 ч/нед)
- Практические работы (по индивидуальной теме) - 15 часов (1ч/нед)
- Проектная работа СРС/СРСП (команды из 2-3 студентов)
- Первая промежуточная аттестация (MidTerm) - на 8й неделе
- Вторая промежуточная аттестация (EndTerm) - на 15й неделе
- Экзамен – Комплексный

# Цель и задачи курса

- Цель курса — дать обучающимся знания и навыки проектирования баз данных, работы с языком структурированных запросов, навыки работы с системами управления базами данных.
- **Задачи:**
- изучение основных концепций баз данных и терминологии;
- умение определить потребности в информации внутри организации и сформулировать пользовательские и организационные требования к базе данных;
- построение концептуальной модели, которая удовлетворяет этим потребностям и требованиям;
- изучение процесса нормализации;
- реализация соответствующей логической модели в системе управления реляционными базами данных (например, PostgreSQL, MySQL и т. д.);
- изучение основ SQL для создания различных запросов для удовлетворения информационных потребностей пользователей.

# Результаты обучения

- **Знать:**
  - Основные концепции баз данных: структуры данных, модели данных, отношения.
  - Принципы проектирования баз данных: нормализация, ключи, отношения между таблицами.
  - Концептуальная, логическая и физическая модели данных;
  - Язык структурированных запросов (SQL) и его основные операторы.
  - Основные типы баз данных: реляционные, NoSQL, их преимущества и недостатки.
- **Уметь:**
  - Создание и обслуживание баз данных с использованием популярных систем управления базами данных (СУБД), таких как MySQL, PostgreSQL, Microsoft SQL Server и другие.
  - Написание эффективных SQL-запросов для извлечения, изменения и управления данными в базах данных.
  - Проектирование баз данных с учетом требований приложения.
  - Работа с транзакциями и обеспечение целостности данных.
- **Владеть навыками:**
  - Разработка баз данных для конкретных приложений и задач.
  - Работа в команде по разработке и сопровождению баз данных.

# Литература

**Томас М. Коннолли**, Системы баз данных: практический подход к проектированию, внедрению и управлению / Томас М. Коннолли, Кэролин Э. Бегг.- Соединенные Штаты Америки: Pearson Education, 2014 (Connolly, Thomas M. Database Systems: A Practical Approach to Design, Implementation, and Management / Thomas M. Connolly, Carolyn E. Begg.- United States of America: Pearson Education, 6th Edition, 2014)

**Ульман, Джеффри**. Реляционные базы данных / Дж. Д. Ульман, Дж. Уидом ; [пер. с англ. П. Быстрова ; науч. ред. В. Алеев] .— Москва : Лори, 2014 .— 374 с. : ил.;

**Garcia-Molina, H.** Database system: The Complete Book / Hector Garcia-Molina.-United States of America: Pearson Prentice Hall

**Sharma, N.** Database Fundamentals: A book for the community by the community / NeerajSharma, LiviuPerniu.-Canada

**Coursera:**

**Databases for Data Scientists Specialization.** Университет Колорадо. Специализация - серия из 3 курсов:

- 1) Relational Database Design (<https://www.coursera.org/programs/masters-management-information-systems-ie9ft/learn/relational-database-design?specialization=databases-for-data-scientists>)
- 2) The Structured Query Language (SQL) (<https://www.coursera.org/programs/masters-management-information-systems-ie9ft/learn/the-structured-query-language-sql?specialization=databases-for-data-scientists>)
- 3) Advanced Topics and Future Trends in Database Technologies (<https://www.coursera.org/programs/masters-management-information-systems-ie9ft/learn/advanced-topics-future-trends-database-technologies?specialization=databases-for-data-scientists>)

## **К концу этого курса слушатели смогут:**

- Проектировать базу данных, начиная с первого этапа и заканчивая физической реализацией
- Продемонстрировать навыки программирования на языке SQL (язык структурированных запросов)
- Работать с системой управления базами данных (СУБД)

Какая у вас ассоциация?



# Что такое база данных?

- База данных представляет собой организованный набор данных, хранящиеся и доступные в электронном виде.
- База данных — это упорядоченное хранение информации в систематизированном виде.



# Пример

- База данных УНИВЕРСИТЕТА
- Информация о студентах, курсах и оценках в университетской среде
- **Записи данных**
- СТУДЕНТ
- КУРС
- ДИСЦИПЛИНЫ
- ЖУРНАЛ ОЦЕНОК

# СУБД – системы управления базами данных

- Общее управление БД осуществляется специально предназначенной для этого системой управления БД, состоящей из языковых, алгоритмических и программных средств.
- Система управления базой данных (СУБД) – это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.



# Функции СУБД

- СУБД, как программное обеспечение, позволяет:
  - создавать, редактировать и управлять БД;
  - работать с БД одновременно несколькими пользователями;
  - упростить работу с базами данных, включая отчетности, резервное копирование, базовые операции с БД и т.п.
- 
- Можно сказать, что основная функция СУБД – это предоставление пользователю БД возможности работы с ней, не вникая в детали на уровне аппаратного обеспечения. То есть все запросы пользователя к БД, добавление и удаление данных, выборки, обновление данных – все это обеспечивает СУБД.

# Приложения

- Программы, с помощью которых пользователи работают с базой данных, называются **приложениями**. Например, если база данных моделирует некоторое предприятие, то для работы с ней может быть создано приложение, которое обслуживает подсистему учета кадров, другое приложение может использоваться для расчета заработной платы сотрудников, третье предназначено для планирования производственного процесса и т. д.
- Приложения могут создаваться как в среде СУБД, так и вне СУБД – с помощью системы программирования, использующей средства доступа к БД (например, Delphi или C++ Builder).
- Для работы с базой данных во многих случаях можно обойтись только средствами СУБД, скажем, создавая запросы и отчеты. Приложения разрабатывают главным образом в случаях, когда требуется обеспечить удобство работы с БД неквалифицированным пользователям или интерфейс СУБД не устраивает пользователя.

# Архитектура базы данных

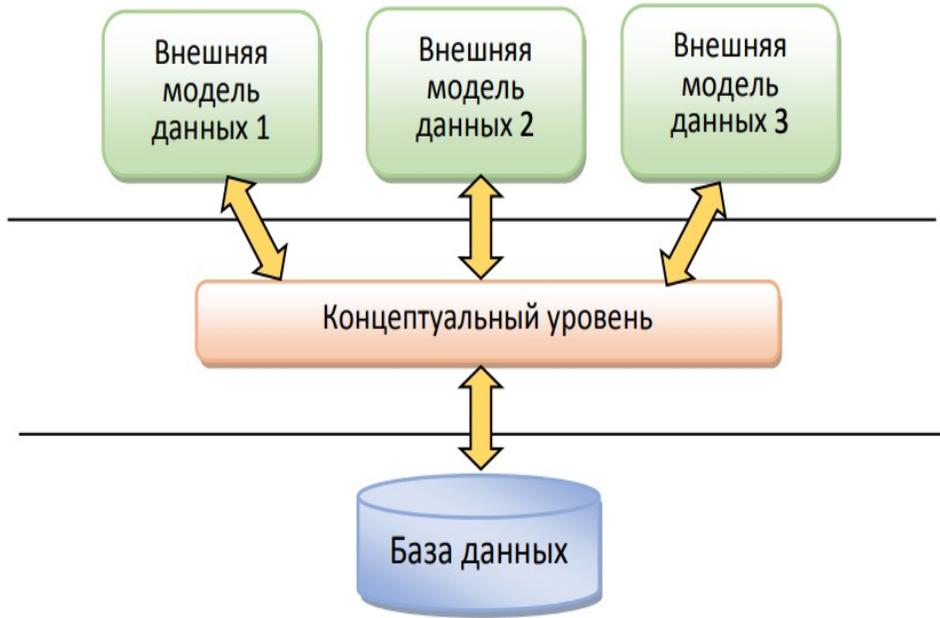


Рис. 1. Трехуровневая система организации БД

**Внешний уровень** Является самым верхним уровнем или уровнем пользователя. Это совокупность внешних представлений данных, которые обрабатывают приложения и какими их видит пользователь на экране. Это может быть таблица с отсортированными данными, с примененным фильтром, форма, отчет, результат запроса.

**Концептуальный уровень** Является центральным. Здесь БД представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями. Фактически концептуальный уровень отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась БД.

**Физический уровень** Это собственно данные, расположенные на внешних носителях.

# Модели баз данных

- Основная задача проектирования базы данных состоит в устранении необходимости переделывания созданной структуры по мере развития системы. Для ее решения создается комплекс взаимосвязанных **моделей данных**.
- **Модель данных** – это некоторая абстракция, которая будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, т.е. сведения, содержащие не только данные, но и взаимосвязь между ними.
- Модель данных должна быть разработана таким образом, чтобы по возможности быть стабильной.

# Модели баз данных

- Известны три основные модели данных:
- **Иерархическая модель** предполагает организацию данных в виде древовидной структуры. На самом верхнем уровне структуры находится корень дерева, не имеющий вышестоящих узлов. Остальные узлы связаны между собой через исходный узел, находящийся выше (рис. 2).

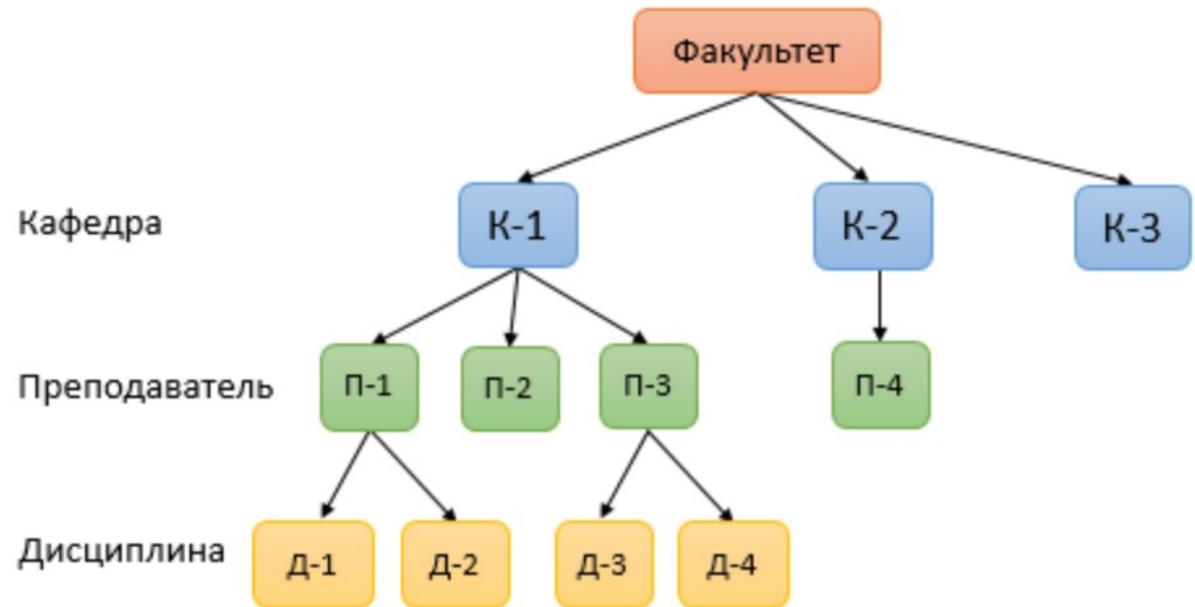


Рис. 2. Пример иерархической структуры данных

# Модели баз данных

- **Сетевая модель** предполагает организацию данных в виде сетевой структуры, когда любой элемент может быть связан с любым другим элементом (рис. 3).

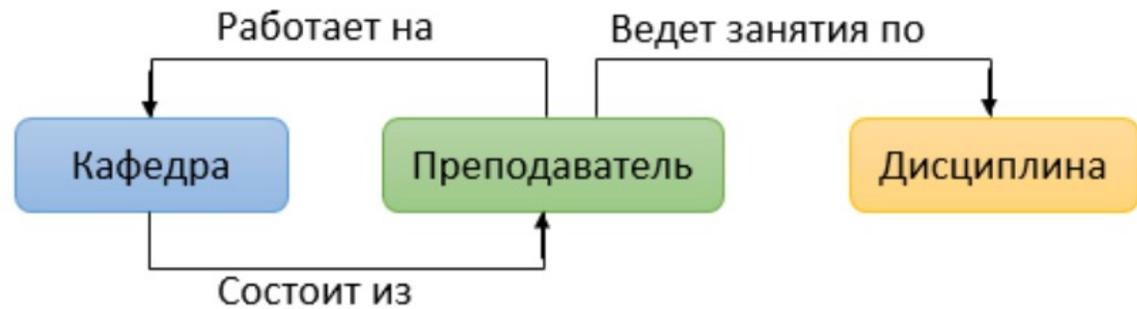


Рис. 3. Пример сетевой модели данных

# Модели баз данных

- **Реляционная модель** (от слова relation – отношение) предполагает использование двумерных таблиц (отношений), связь между которыми осуществляется посредством значений одного или нескольких совпадающих полей.
- При этом каждая строка таблицы уникальна, что обеспечивается использованием ключей, содержащих одно или несколько полей таблицы.

Факультет

Код_фак	Название_фак
10	Математический
20	Физический
30	Исторический

Кафедра

Код_фак	Код_каф	Название_каф
10	1	Прикладной математики и кибернетики
10	2	Математического анализа
10	3	Геометрии и топологии
20	4	Электроники
20	5	Общей физики

Рис. 4. Пример реляционной модели данных

# Реляционная баз данных

- Основным объектом **реляционных баз данных** является **таблица**. Простейшая база данных имеет хотя бы одну таблицу. Структуру любой двумерной таблицы составляют столбцы и строки, аналогами которых в базе данных являются поля и записи (рис. 5).

**Поле (столбцы)** – это элементарная единица логической организации данных, которая соответствует неделимой единице информации (атрибуту)

**Запись (строки, кортеж)** – это совокупность логически связанных полей

**Таблица (отношение)** – это совокупность записей одной структуры.

как правило, каждая **таблица** представляет одну сущность (например, Студенты).

- строки** представляют экземпляры этого типа объект (например, студент1, студент2 и т. д.)
- столбцы**, представляющие атрибуты, экземпляра (например, stud\_id, last\_name, так далее.)

Код	Название	Группа	Высота	Диаметр	USDA
1	Ангела	флорибунда	100-150	5-6	5-9
2	Вестерланд	шрабы	160-200	10-11	5-9
3	Индиголетта	плетистые	250-300	8-10	6-9

Рис. 5. Пример реляционной БД

# Элементы реляционных баз данных



Рис. 5. Пример реляционной БД

В структуре записи указываются **ключевые поля**, которые могут быть простыми или составными. Одно или несколько полей, комбинация значений которых однозначно определяет каждую запись в таблице, называется **первичным (главным) ключом**. При этом в таблице не может быть одинаковых первичных ключей. Поле «Код» в примере, приведенном на рис. 5, однозначно определяет запись и является первичным ключом. Он является простым, так как состоит из одного поля.

# Этапы проектирования базы данных

- 1. Анализ предметной области
- 2. Концептуальное проектирование
- 3. Логическое проектирование
- 4. Физическое проектирование

# Анализ предметной области

- Первым этапом проектирования БД любого типа является **анализ предметной области**, который заканчивается построением информационной структуры (концептуальной схемы).
- На данном этапе анализируются запросы пользователей, выбираются информационные объекты и их характеристики, которые определяют содержание проектируемой БД.
- На основе проведенного анализа структурируется предметная область. Анализ предметной области не зависит от программной и технической сред, в которых будет реализовываться БД.

# Анализ предметной области

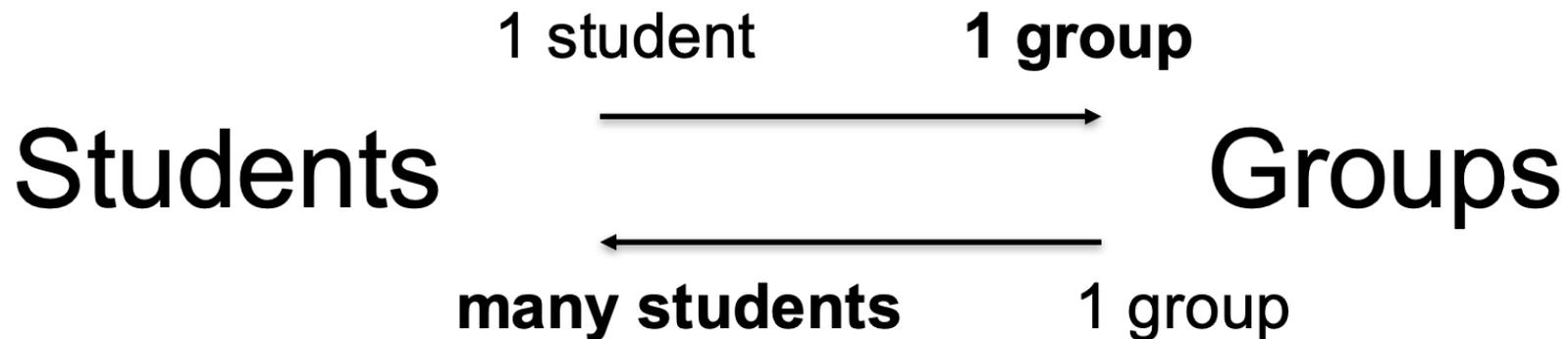
- **1. Указать какие данные будут храниться в будущей БД:**  
«БД предназначена для хранения информации, касающейся/о...»
- **2. Перечислите сущности и атрибуты** (проще говоря, таблицы и столбцы будущей базы данных). Каждая сущность должна иметь уникальный атрибут (Первичный ключ)

# Пример

- Описание таблицы — это имя таблицы и его атрибуты:
- Студенты (`stud_id, last_name, bdate, phone`)
- **Первичный ключ** — это атрибут с уникальными значениями — подчеркните его.
- Студенты (`stud_id`, `last_name, bdate, phone`)

# Анализ предметной области

- **3. Связь между сущностями**
- Чтобы описать связь между двумя сущностями (таблицами), ответьте на 2 вопроса. **Первый** из них — как одна строка из первой таблицы связана с одной или несколькими строкам из второй таблицы, а **второй** вопрос (с другой стороны) — как одна строка из второй таблицы связана с одной или несколькими строкам из первой таблицы.



# Анализ предметной области

- **4. Ограничения (Constraints).**
- Пример: Дата рождения студента должна быть не ранее 1980 года.
  
- **5. Укажите группы пользователей и их права доступа:** «База данных предназначена для ...»
  
- **6. Перечислите потенциальные вопросы пользователей к базе данных (запросы)**
- Пример: Найдите имя студента по его коду.

Спасибо за внимание!