



БАЗЫ ДАННЫХ

Петелина Юлия Павловна

Ассистент кафедры ИОИД ФИТ

Современные информационные системы содержат:

- Аппаратное обеспечение,
- Программное обеспечение,
- Системы управления базами данных,
- Другие технические средства хранения, обработки и передачи информации.

Компьютерные информационные системы позволяют хранить большие объемы данных, осуществлять в них быстрый поиск, вносить изменения, выполнять всевозможные манипуляции

Модели данных

Совокупность правил, в соответствии с которыми должны быть организованы данные и допустимые операции с ними, называется **моделью данных**.

•иерархическая

•сетевая

•реляционная

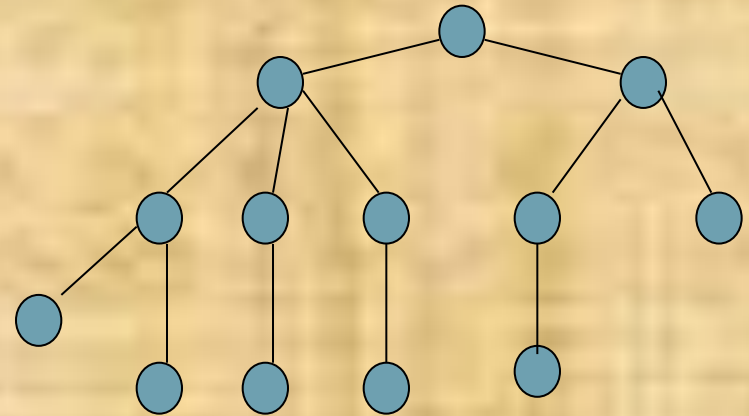
Модель данных определяется прежде всего *ПРИРОДОЙ* объекта, поставляющего эти данные. Однако в некоторых случаях, когда это не задевает принципиальные особенности объекта, можно взять более простую модель данных, простую в смысле возможности автоматической обработки с помощью ЭВМ.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ

представляется собой данные в виде древовидной структуры, состоящей из данных различных уровней.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня.

Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок.



В иерархической модели каждый внутренний узел связан только с одним исходным, а порождает несколько последующих.

ПРИМЕР ИЕРАРХИЧЕСКОЙ МОДЕЛИ ДАННЫХ

Уровень 1

Школа

Уровень 2

Первые классы

Вторые классы

Третьи классы

...

Уровень 3

1 «А»

1 «Б»

2 «А»

2 «Б»

3 «А»

3 «Б»

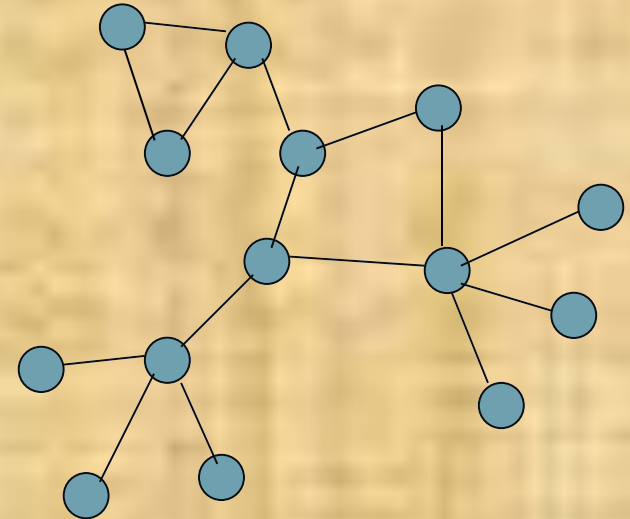
Уровень 4

Отдельные ученики разных классов

СЕТЕВАЯ МОДЕЛЬ

Сетевая модель данных отличается от иерархической тем, что любой её элемент может быть связан с любым другим элементом, пример - глобальная сеть интернет.

В сетевой структуре данных у потомка может иметься любое число предков.



В сетевой модели принята свободная связь элементов между собой.

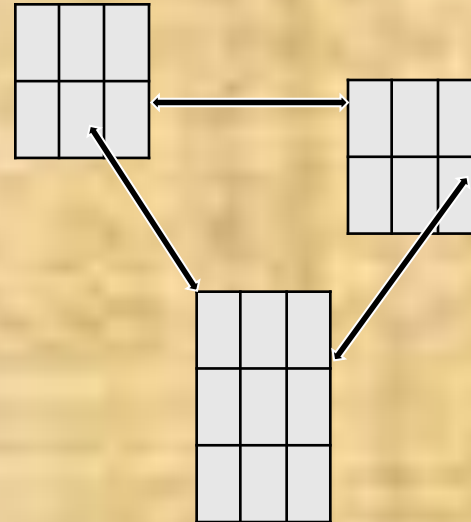
ПРИМЕР СЕТЕВОЙ МОДЕЛИ ДАННЫХ



Таким образом, как иерархическая, так и сетевая модели данных состоят из совокупности *узлов* и *связей* между ними. Каждый узел (элемент данных) в обеих моделях является *записью*.

РЕЛЯЦИОННАЯ МОДЕЛЬ

Реляционная модель представляет собой набор взаимосвязанных таблиц (отношений).



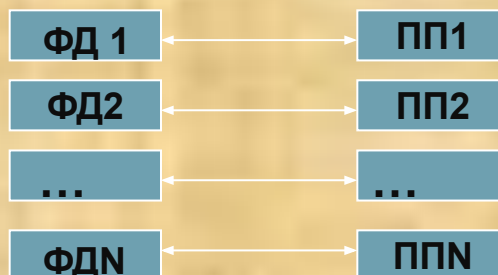
Элемент реляционной модели данных имеет вид таблицы, которая содержит конечное множество однотипных записей.

ПРИМЕР РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ

№ ЗАЧЕТНОЙ КНИЖКИ	ФИО	ДАТА РОЖДЕ- НИЯ	КУРС	СПЕЦИ- АЛЬНОСТЬ
21316	Иванов А.В.	23.11.85	2	биология
22121	Петров Г.С.	17.02.87	1	история
20118	Сидоров К.Т.	03.10.84	3	биология
21208	Волков А.А.	11.03.85	2	физика
19950	Антонов С.А.	02.06.83	4	экономика

СПОСОБЫ ОРГАНИЗАЦИИ ДАННЫХ

файловый подход



Преимущества файлового подхода:

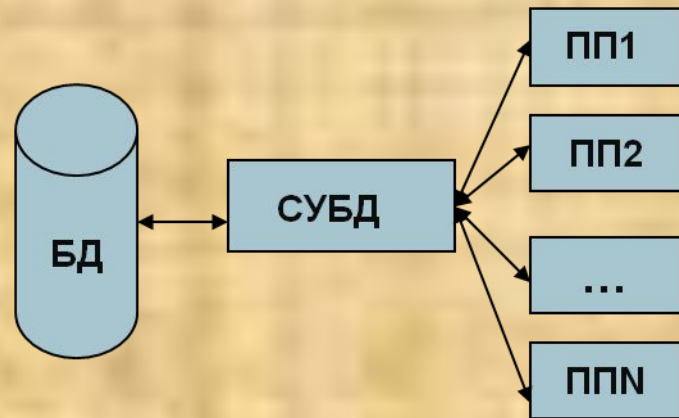
- разделение данных и программ их обработки по субъекту пользования;
- высокая скорость обработки данных за счёт тесной связи с программами.

Недостатки файлового подхода:

- избыточность: данные в различных блоках дублируются;
- изменения в организации данных требуют коррекции программ обработки.

Концепция БД

База данных (БД) – именованная совокупность данных, отображающих состояние объектов и их отношения в рассматриваемой предметной области.



Преимущества баз данных:

- интеграция данных: для всех программ пользователей данные накапливаются, хранятся и обновляются централизованно;
- обеспечение логической и физической зависимости данных;
- отсутствие проблемы контроля избыточности данных в силу их интеграции.

БД – это совокупность данных, которой свойственны:

- 1) интегрированность: данные направлены на решение общих задач;
- 2) модельность: данные представляют объекты реального мира;
- 3) взаимосвязанность;
- 4) данные и их описания хранятся совместно в БД и не зависят от логики программ пользователей.

Блок данных, отвечающий требованиям 1)-4), можно рассматривать в качестве базы данных.

СУБД

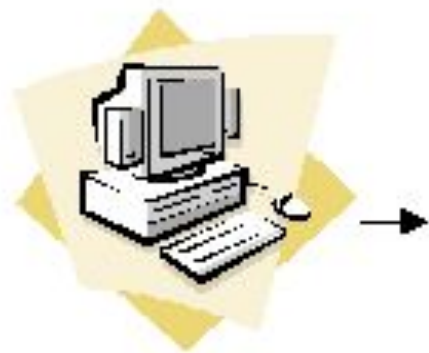
Система управления базами данных (СУБД) — комплекс программных средств для создания баз данных, хранения и поиска в них необходимой информации.

Основные функции СУБД:

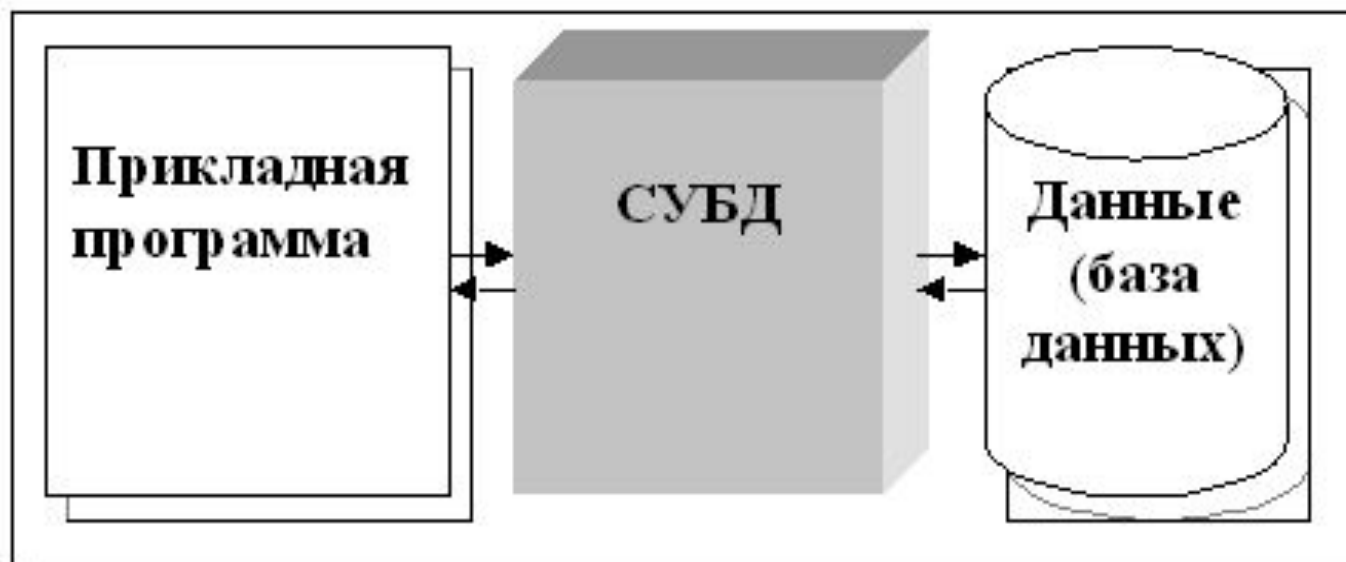
- 1) поддержка выбранной пользователем модели данных;
- 2) использование языковых средств описания и манипуляции данными SQL (Structured Query Language);
- 3) поддержка операций создания и обработки данных (обновление, удаление, вставка, выбор по запросу и т.д.);
- 4) защита и целостность данных при коллективном пользовании.

Некоторые СУБД

Access, **FoxPro**, Paradox, **dBase**, FoxBase, **Oracle**, **Postgres**, Typhoon, LINCKS, Exodus, InterBase, PostgreSQL, Firebird, **MySQL**, Microsoft SQL, IBM DB2, Sav Zigzag, SQLite, Apache CouchDB, Cache, XSForms, eSQL ...



Персональный
компьютер или
мэйнфрейм



Классификация БД

По характеру использования БД делят на:

- **персональные;**
- **многопользовательские (корпоративные).**

По технологии обработки данных БД подразделяются на:

- **централизованные** (хранится в памяти одной ЭВМ);
- **распределенные** (состоит из нескольких частей, хранящихся на различных ЭВМ вычислительной сети).

По способу доступа к данным БД разделяются на:

- БД с **локальным** доступом (БД размещена на одной отдельной ЭВМ и считывание информации из БД производится непосредственно с этой машины);
- БД с **удаленным** (сетевым) доступом (управление машиной, на которой размещена БД, производится по сети с другой ЭВМ).

По способу установления связей между данными различают:

реляционные, иерархические и сетевые БД. 15

Проектирование баз

данных
Проектирование БД представляет собой длительный и трудоемкий процесс и является скорее искусством, чем наукой. Основными ресурсами проектировщика БД служат его собственные интуиция и опыт.

Уровни проектирования:

- 1) Концептуальный.** На естественном языке с помощью диаграмм и других средств описываются объекты предметной области и их взаимосвязи.
- 2) Логический.** Выбирается модель данных (сетевая, иерархическая, реляционная), производится отображение данных концептуальной модели в логическую модель в рамках выбранной модели данных.
- 3) Физический.** Производится выбор СУБД, типов данных и методов доступа к ним, которые обеспечивает выбранная СУБД.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ БД

Одним из наиболее удобных инструментов представления той информации, которую предполагается хранить в создаваемой БД, является инфологическое моделирование (или модель «сущность-связь»).

Представление данных с помощью модели «сущность-связь»

Основными конструктивными элементами инфологических моделей являются *сущности, связи между ними и их свойства (атрибуты)*.

Сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть *люди, предметы, события*.

Экземпляр сущности – конкретный представитель сущности. Например, сущность ГОРОД, экземпляры – Москва, Киев и т.д.

Атрибут – поименованная характеристика сущности. Примерами атрибутов для сущности АВТОМОБИЛЬ являются ТИП, МАРКА, НОМЕРНОЙ ЗНАК, ЦВЕТ и т.д. для сущности СОТРУДНИК атрибутами являются ТАБЕЛЬНЫЙ_НОМЕР, ФАМИЛИЯ, ВОЗРАСТ и др.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

Например, ключом сущности АВТОМОБИЛЬ является НОМЕРНОЙ ЗНАК.

Связь – это ассоциация, установленная между несколькими сущностями.

Например,
поскольку каждый сотрудник работает в каком-либо отделе, между сущностями СОТРУДНИК и ОТДЕЛ существует связь, которую назовем «работает в»;

могут существовать и связи между сущностями одного типа, например связь «родитель – потомок» между двумя сущностями ЧЕЛОВЕК;


Связь также может иметь атрибуты.

Например, для связи «работает в» можно задать атрибут СТАЖ_РАБОТЫ_В_ОТДЕЛЕ

Язык моделирования

При построении инфологических моделей можно использовать язык диаграммы «сущность-связь» (ER-диаграммы), которые представляют собой графические схемы.

Основные обозначения, используемые в ER-диаграммах:

Обозначение	Значение
 имя сущности	сущность
 имя атрибута	атрибут
 <u>имя атрибута</u>	ключевой атрибут
 имя связи	связь
	прямыми линиями соединяются атрибуты с сущностями и сущности со связями

Пример ER-диаграммы:



Характеристика связей

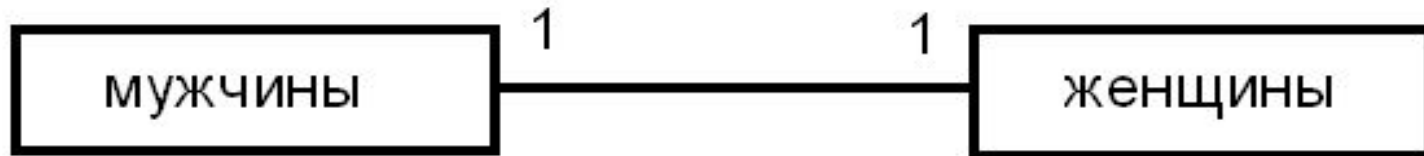
Между двумя сущностями, например, А и В возможны следующие виды связей.

1. Связь **ОДИН-К-ОДНОМУ (1:1)**: в каждый момент времени каждому представителю (экземпляру) сущности А соответствует 1 или 0 представителей сущности В.

ПРИМЕР изображения связи на ER-диаграмме:



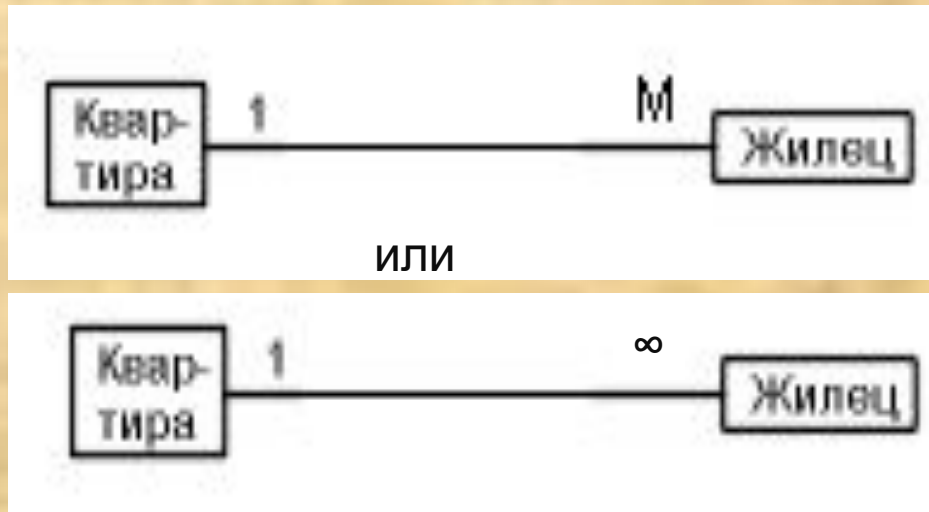
или



Мужчина женат на одной женщине (традиционный брак).

2. Связь **ОДИН-КО-МНОГИМ (1:М)**: одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В.

ПРИМЕР:



Квартира может пустовать, в ней может жить один или несколько жильцов.

Между двумя сущностями возможны связи в обоих направлениях, поэтому существует связь **МНОГИЕ-К-ОДНОМУ (M:1)**

3. Связь **МНОГИЕ-КО-МНОГИМ (M:N)**: одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В, а одному представителю сущности В соответствуют 0, 1 или несколько представителей сущности А.

ПРИМЕР:



Покупатель может приобретать товар у нескольких продавцов, продавец может продавать товар нескольким покупателям

Более сложные связи:

- множество связей между одними и теми же сущностями



Пациент, имея одного лечащего врача, может иметь также несколько врачей-консультантов; врач может быть лечащим врачом нескольких пациентов и может одновременно консультировать несколько других пациентов;

- тернарные связи



Врач может назначить несколько пациентов на несколько анализов, анализ может быть назначен несколькими врачами нескольким пациентам и пациент может быть назначен на несколько анализов несколькими врачами.

связи более высоких порядков (смысл которых иногда очень сложен)

ЛОГИЧЕСКАЯ МОДЕЛЬ БД

- Это создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Для построения логической модели требуется выполнить следующие действия:

1. Создать по одной таблице для каждой сущности и каждой связи.
2. Для каждой таблицы задать первичный и внешний ключи

ФИЗИЧЕСКАЯ МОДЕЛЬ БД

В физической модели содержится информация обо всех объектах БД (таблицах, индексах, процедурах и др.) и используемых типах данных.

Физическая модель зависит от конкретной СУБД. Одной и той же логической модели может соответствовать несколько разных физических моделей.

Физическое проектирование является начальным этапом реализации БД

Пример проектирования базы данных

Разработать базу данных, в которой будут храниться личные данные студентов (фамилия, год рождения, адрес), изучаемые дисциплины, оценки, данные о преподавателях (фамилия, наименование кафедры, должность, читаемая дисциплина).

1 этап Разработка концептуальной модели БД

Первый этап проектирования заключается в описании объектов базы данных – сущностей, определении их характеристик (атрибутов) и в установлении связей между сущностями.

Для проектируемой базы данных можно выделить три объекта (сущности), которые не будут обладать избыточностью:

- *Студент,*

-*Преподаватель,*

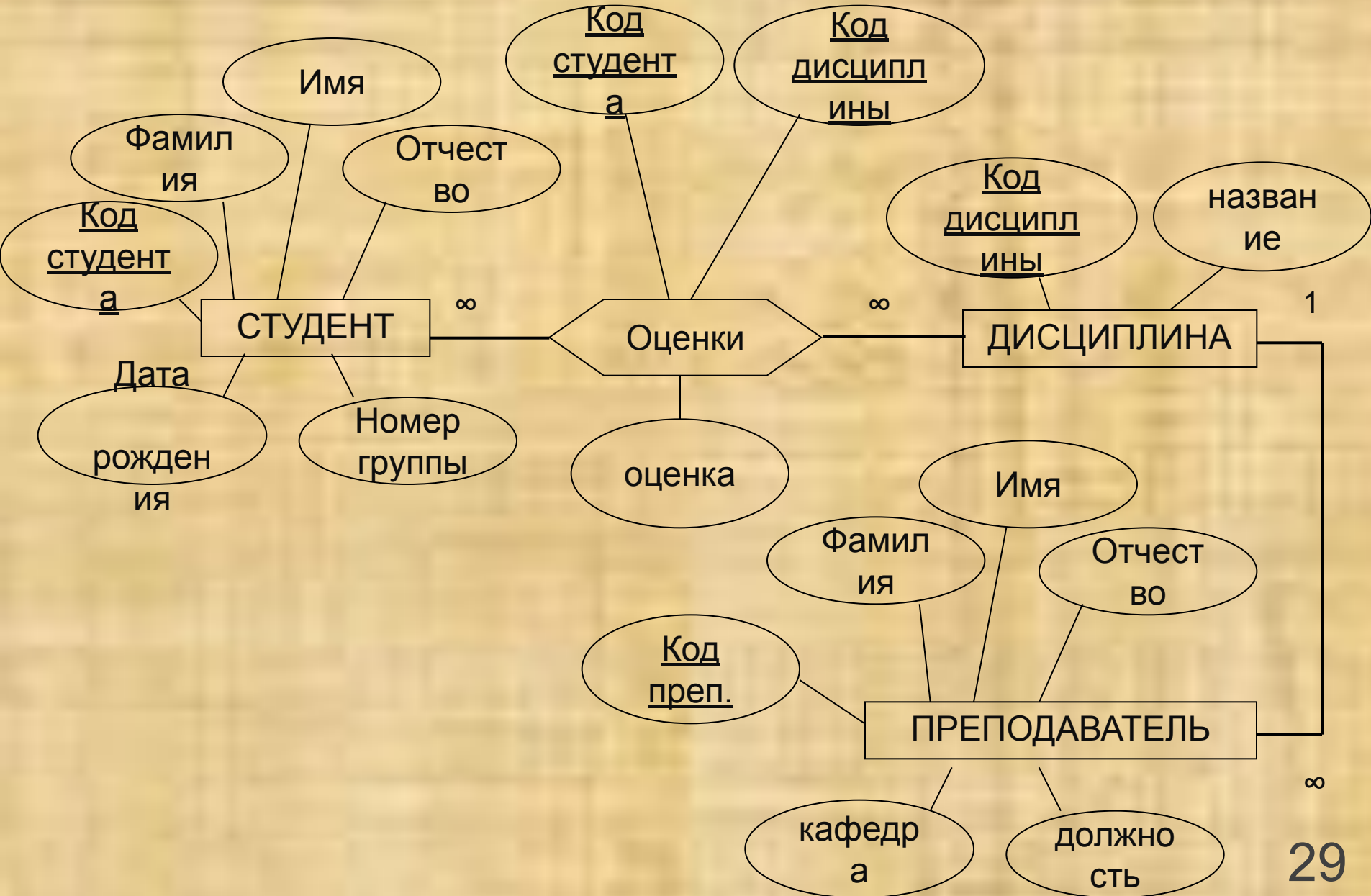
- *Дисциплина.*

Студент	
Код студента	Число
Фамилия	Текст
Имя	Текст
Отчество	Текст
Дата рождения	Дата
Номер группы	Текст

Дисциплина	
Код дисциплины	Число
Название дисциплины	Текст

Преподаватель	
Код преподавателя	Число
Фамилия	Текст
Имя	Текст
Отчество	Текст
Кафедра	Текст
Должность	Текст

Концептуальная модель БД (ER-диаграмма)



2 этап Разработка логической модели БД

1. Создать по одной таблице для каждой сущности и связи, имеющей атрибуты.
2. Для каждой сущности задать первичный и внешний ключи.



3 этап Разработка физической модели БД

Этот этап представляет реализацию логической модели с помощью конкретной СУБД, например Access.

Имя поля	Тип данных	Описание
Код студента	Счетчик	
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Номер группы	Числовой	
Дата рождения	Дата/время	
Стипендия	Числовой	

Свойства поля

Общие Подстановка

Размер поля: Длинное целое
Новые значения: Последовательные
Формат поля:
Подпись:
Индексированное поле: Да (Совпадения не допускаются)
Смарт-теги:

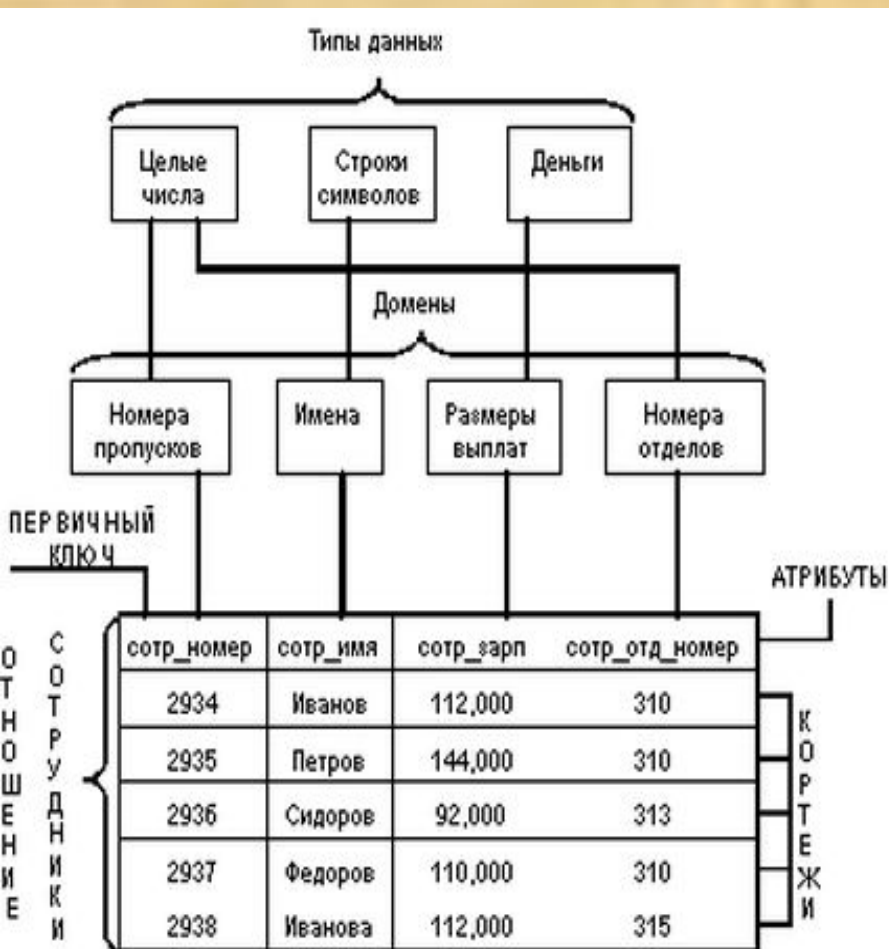
Тип данных определяет значения, которые можно сохранять в этом поле. Для справки по типам данных нажмите клавишу F1.

Создание таблиц с помощью **Конструктора**

Теоретические основы реляционных баз данных

Атрибут, столбец, поле – имя каждого из столбцов таблицы.

Кортеж (запись, строка, экземпляр) – это элемент отношения, представляет собой набор из n -значений, по одному значению для каждого атрибута отношения (n – количество атрибутов).



Домен – множество допустимых значений для заданного атрибута.

Не все они обязательно присутствуют в данной таблице, а являются *возможными в принципе*.

Например, домен “Имена” – определен на базовом типе строк (символов), но в число его значений могут входить только те строки, которые представляют имена.

Различные атрибуты могут быть определены на одном домене, например, дата рождения и дата приема на работу.

Ключ отношения, первичный ключ или ключевое поле – это один из атрибутов в данном отношении, который однозначно определяет (идентифицирует) каждую запись в таблице.

То есть, значения ключевого атрибута для всех экземпляров данной таблицы различны. Ключ отношения для удобства размещают в первом столбце.

Примеры ключа: № зачетной книжки, ИНН налогоплательщика.

Помимо простого ключа (№ лицевого счета в банке) может встречаться **составной ключ** (состоять из нескольких атрибутов, например, ПАСПОРТНЫЕ ДАННЫЕ: СЕРИЯ + № ПАСПОРТА)

Схема отношения представляется:

1. Перечнем имен отношения и атрибутов в виде $R(A_1, A_2, \dots, A_k)$, где R – имя отношения; A_1, \dots, A_k – имена атрибутов.
2. Описанием доменов для каждого атрибута
3. Условиями ограничения целостности, задаваемых на доменах (например, дата рождения не должна определять возраст более 200 лет).

Степень отношения – число его атрибутов (равна 4 в примере).

Кардинальное число – количество записей в таблице.

Признаки отношения (необходимые):

- Шапка таблицы состоит из конечное множества атомарных атрибутов (нет подзаголовков).
- Каждая таблица имеет имя и первичный ключ.
- Все атрибуты различны, а их значения однотипны.
- Для каждого значения ключа существует запись причем только одна (ссылочная целостность).
- Изменение значения в любом неключевом поле не должно изменять значение другого поля (функциональная независимость неключевых атрибутов)

Резюме: выполнение 1)- 5) позволяет считать таблицу отношением, то есть объектом реляционных операций.

Пример связного отношения

студент

фио	курс	специальность
Иванов	2	химия
Петров	4	бух. учёт
Титов	3	физика

предмет

название	число семестров
алгебра	4
история	3
информатика	4
иностр. язык	2

изучает

студент	предмет
Иванов	алгебра
Иванов	информатика
Петров	иностр. язык
Титов	история

Для двух объектных отношений
СТУДЕНТ(ФИО, КУРС, СПЕЦИАЛЬНОСТЬ)
ПРЕДМЕТ(НАЗВАНИЕ, ЧИСЛО СЕМЕСТРОВ)
связным отношением может быть
ИЗУЧАЕТ(СТУДЕНТ, ПРЕДМЕТ), то есть оно
содержит ключи обоих объектных отношений.

Нормализация отношений

Важным этапом создания реляционной базы данных является **нормализация отношения**.

Под **нормализацией** отношения подразумевается процесс декомпозиции (разбиения) исходных отношений БД на другие, более мелкие и простые отношения. При этом устанавливаются все возможные функциональные зависимости.

Зачем нужна нормализация

Для обеспечения достоверности и непротиворечивости хранимых данных каждый факт должен храниться в БД один-единственный раз. Причем эти свойства не должны утрачиваться в процессе работы с данными.

Для поддержания БД в таком устойчивом состоянии используется ряд механизмов, которые получили обобщенное название **средств поддержки целостности**. К таким средствам, в частности, относится и нормализация.

Всего в реляционной теории насчитывается 6 НФ:

1. Первая нормальная форма (обычно обозначается 1НФ).
2. 2НФ
3. 3НФ
4. Нормальная форма Бойса-Кодда (НФБК).
5. 4НФ.
6. 5НФ.

На практике, как правило, ограничиваются третьей нормальной формой, ее оказывается вполне достаточно для создания надежной схемы БД.

Первая нормальная форма (1НФ)

Простой атрибут - атрибут, значения которого атомарны (неделимы).

Сложный атрибут - получается соединением нескольких атомарных атрибутов.

Отношение находится в **первой нормальной форме**, если все его атрибуты простые.

Это означает, что каждый атрибут отношения должен хранить одно-единственное значение и не являться ни списком, ни множеством значений.

Приведение отношения к 1НФ

Для этого необходимо просмотреть схему отношения и разделить составные атрибуты на различные строки/столбцы. Повторить эту операцию несколько раз до тех пор, пока каждый из атрибутов не станет атомарным.

Пример.

СТУДЕНТ

Фамилия	Курс	Специальность	Спорт	
			Вид	Разряд
Иванов	2	Математика	плавание	м. с.
Петров	4	физика	футбол	к. м.с.
Сидоров	3	экономика	шахматы	1 разряд

Очевидно, что в данном случае атрибут "Спорт" является сложным.

Приведем это отношение к 1НФ, то есть избавимся от сложного атрибута:

СТУДЕНТ

Фамилия	Вид спорта	Курс	Специальность	Спорт_разряд
Иванов	плавание	2	математика	м. с.
Петров	футбол	4	физика	к. м.с.
Сидоров	шахматы	3	экономика	1 разряд

Теперь в отношении СТУДЕНТ все атрибуты простые, следовательно это отношение находится в 1НФ.

Вторая нормальная форма (2НФ)

Определение. Отношение находится во **второй нормальной форме**, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от составного ключа.

Замечание: вышесказанное относится к отношениям с ***составным ключом***. Отношение с простым ключом (состоящим из единственного атрибута), приведенное к 1НФ, находится во 2НФ **по определению** и в данном этапе нормализации не нуждается.

Приведение отношения к 2НФ

Для иллюстрации используем отношение ПРЕПОДАВАТЕЛЬ-ПРЕДМЕТ с составным ключом «Личный номер» и «Название предмета» из предыдущей лекции

ПРЕПОДАВАТЕЛЬ - ПРЕДМЕТ

Личный номер	Название предмета	Кол-во часов	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Информатика	36	Фролов	доцент	3900	ПМ	23-33-15
201	ИТ	24	Фролов	доцент	3900	ПМ	23-33-15
202	Физика	48	Костин	профессор	5600	Физика	23-45-19
401	Экономика	36	Глазов	ассистент	1999	Экономика	23-56-90
401	Бухучет	16	Глазов	ассистент	1999	Экономика	23-56-90

Отметим, что:

Оно находится в 1НФ, поскольку не содержит составных атрибутов.

В этом отношении можно отметить частичную зависимость атрибутов «Фамилия», «Должность», «Оклад», «Кафедра», «Телефон» от части «Личный номер» составного ключа.

Такая частичная зависимость приводит к следующим проблемам:

1. Имеет место **дублирование данных** о преподавателе, поскольку преподаватель может читать несколько предметов.
2. Существует проблема контроля **избыточности данных**, так как изменение, например, оклада влечет за собой необходимость поиска и изменения значений окладов во всех записях с данным преподавателем.
3. Возникает проблема с преподавателями, которые в данное время не ведут предметы, а именно, преподавателя без предмета невозможно включить в отношение и наоборот, если преподаватель увольняется и удаляется из отношения, то будет удален и предмет, хотя предмет должен продолжать читаться.

Личный номер	Название предмета	Кол-во часов	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Информатика	36	Фролов	доцент	3900	ПМ	23-33-15
201	ИТ	24	Фролов	доцент	3900	ПМ	23-33-15
202	Физика	48	Костин	профессор	5600	Физика	23-45-19
401	Экономика	36	Глазов	ассистент	1999	Экономика	23-56-90
401	Бухучет	16	Глазов	ассистент	1999	Экономика	23-56-90

Чтобы устранить частичную зависимость и привести рассматриваемое отношение к 2НФ, необходимо разбить его на два отношения:

В итоге получим два отношения – ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, находящиеся в 2НФ.

ПРЕДМЕТ

ПРЕПОДАВАТЕЛЬ

Личн ый номер	Название предмета	Кол- во часов
201	Информатика	36
201	ИТ	24
202	Физика	48
401	Экономика	36
401	Бухучет	16

Личн ый номер	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Фролов	доцент	3900	ПМ	23-33-15
201	Фролов	доцент	3900	ПМ	23-33-15
202	Костин	профессор	5600	Физика	23-45-19
401	Глазов	ассистент	1999	Экономика	23-56-90
401	Глазов	ассистент	1999	Экономика	23-56-90

Третья нормальная форма (3НФ)

Определение. Отношение находится в **третьей нормальной форме**, если оно находится в 2НФ и в нем отсутствуют транзитивные зависимости неключевых атрибутов от ключа.

В полученном отношении ПРЕПОДАВАТЕЛЬ имеются **транзитивные функциональные зависимости**, например:

Личный номер → Кафедра → Телефон

Личный номер → Должность → Оклад

Наличие транзитивных зависимостей порождает следующие неудобства (на примере атрибута «Телефон»):

- 1. Имеет место дублирование информации о телефоне для преподавателей одной кафедры, это порождает проблему контроля избыточности, поскольку изменение номера телефона кафедры влечет за собой необходимость поиска и изменения номеров всех преподавателей этой кафедры.**
- 2. Нельзя включить данные о новой кафедре (название и номер телефона), если на данный момент еще отсутствуют преподаватели.**

Для решения этих проблем необходимо разбить отношение ПРЕПОДАВАТЕЛЬ на три следующих отношения:

ПРЕПОДАВАТЕЛЬ

Личный номер	Фамилия	Должность	Кафедра
201	Фролов	1	1
202	Костин	2	2
401	Глазов	3	3

КАФЕДРА

Номер	Кафедра	Телефон
1	ПМ	23-33-15
2	Физика	23-45-19
3	Экономика	23-56-90

ДОЛЖНОСТЬ

Номер	Должность	Оклад
1	доцент	3900
2	профессор	5600
3	ассистент	1999

Каждое из этих отношений находится в ЗНФ

3 НФ освобождает от избыточности данных и аномалий выполнения операций включения, удаления и изменения записей.

Резюме: нормализация увеличивает число отношений в БД и тем самым может возрасти время обработки данных. Но благодаря корректности и устранению дублирования данных ускоряется выполнение операций доступа к данным.