

# Реляционные базы данных

---

## ЛЕКЦИЯ 2

ассистент кафедры Интеллектуальных  
информационных технологий УрФУ

Глухих Олег Юрьевич

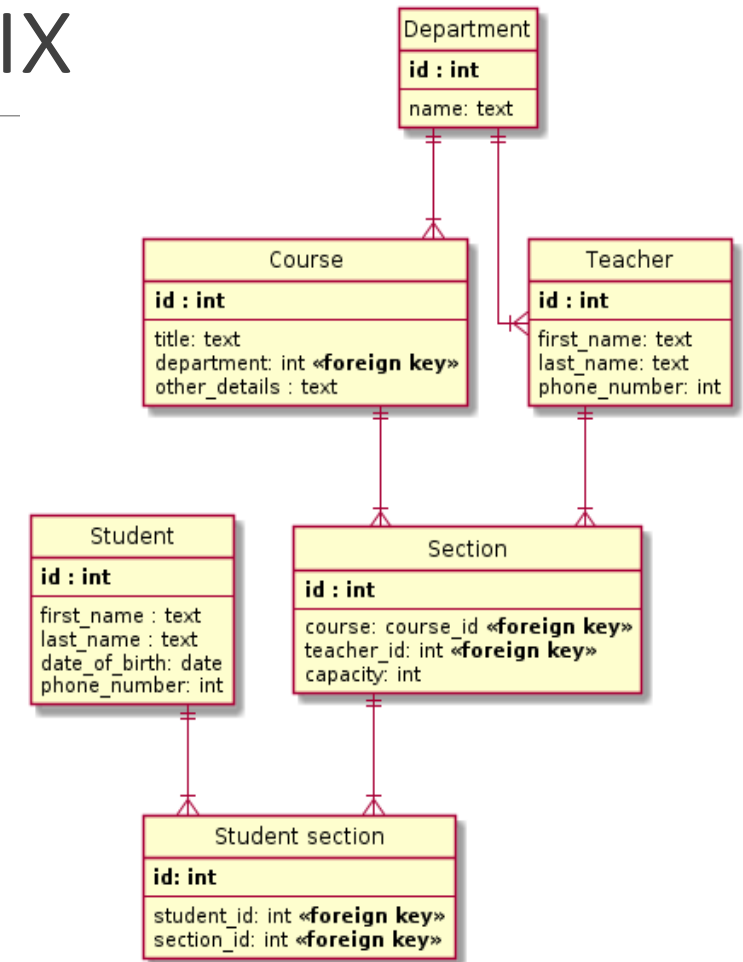
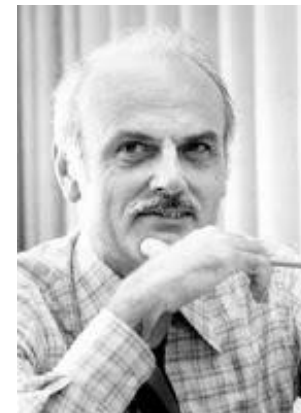
# Реляционная модель данных

Реляционная модель данных основывается на математических принципах, вытекающих непосредственно из теории множеств и логики предикатов. Эти принципы впервые были применены в области моделирования данных в конце 1960-х гг. доктором *Е.Ф. Коддом*, в то время работавшим в IBM, а впервые опубликованы - в 1970 г.

## Эдгар Франк «Тед» Кодд

(23 августа 1923 — 18 апреля 2003)

британский учёный, работы которого заложили основы теории реляционных баз данных. Работая в компании IBM, он создал реляционную модель данных.



# Реляционная модель данных: достоинства и недостатки

---

Табличная организация БД позволяет реализовать ее важнейшее **преимущество** перед другими моделями данных:

- возможность использования точных математических методов манипулирования данными, и, прежде всего, аппарата реляционной алгебры и исчисления отношений,
- наглядность,
- простота изменения данных и организация разграничения доступа к ним.

Основным **недостатком** реляционной модели данных является: информационная избыточность, что ведет, к перерасходу ресурсов вычислительных систем.

Примеры:

- [MySQL](#)
- [MariaDB](#)
- [PostgreSQL](#)
- [SQLite](#)

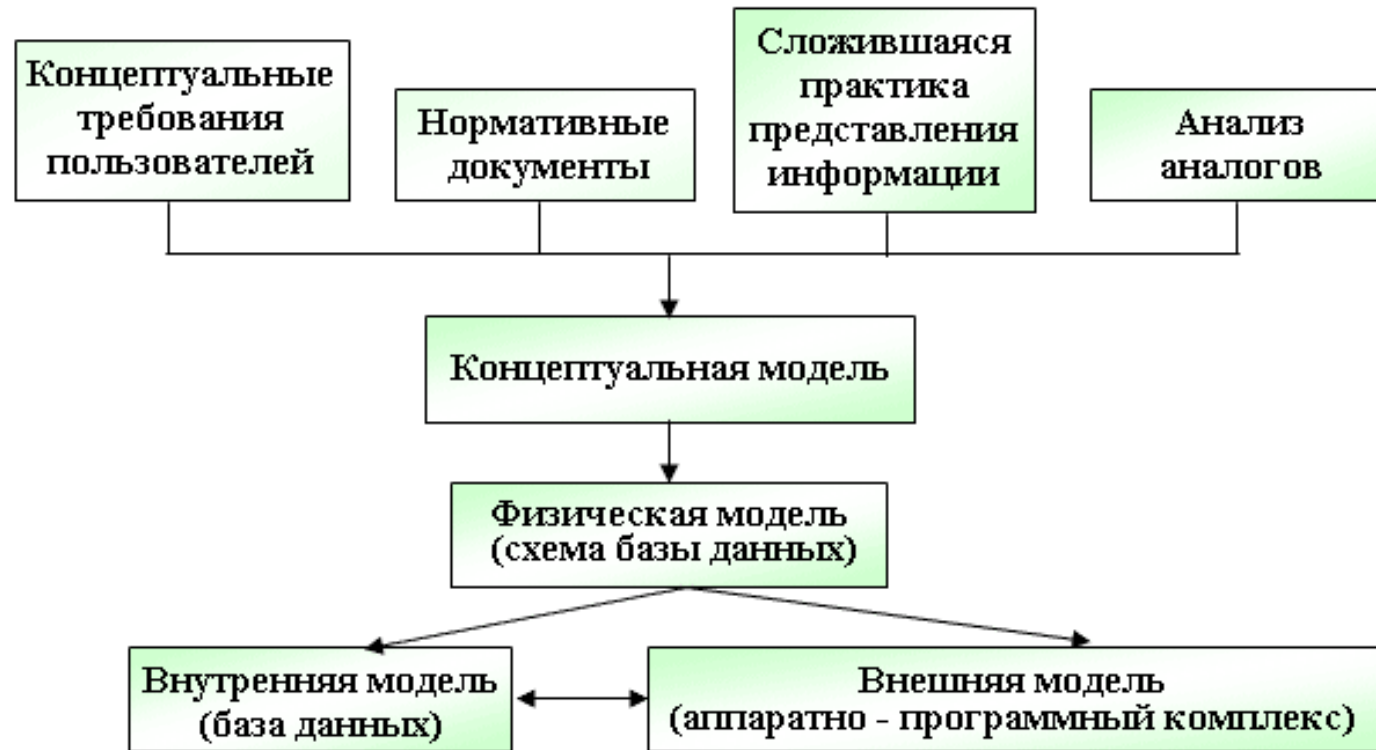
# Термины реляционной модели

---

<b>Термин реляционной модели</b>	<b>Эквивалентный термин</b>
Отношение	Таблица
Схема отношения	Строка заголовков столбцов таблицы (заголовок таблицы)
Кортеж	Строка
Сущность	Описание свойств объекта
Атрибут	Столбец, поле
Домен	Множество допустимых значений атрибута
Первичный ключ	Уникальный идентификатор
Кардинальность	Количество строк
Степень	Количество столбцов

# Схема формирования информационной модели

---



# Концептуальная модель

---

**Концептуальная модель** - отображает информационные объекты, их свойства и связи между ними без указания способов физического хранения информации (модель предметной области, иногда ее также называют информационно-логической или инфологической моделью).

Информационными объектами обычно являются **сущности** - обособленные объекты или события, информацию о которых необходимо сохранять, имеющие определенные наборы свойств - **атрибутов**.

Основные принципы реляционной модели на концептуальном уровне, с точки зрения теории реляционных БД:

---

все данные представляются в виде упорядоченной структуры, определенной в виде строк и столбцов и называемой **отношением**;

все значения являются **скалярами**. Это означает, что для любой строки и столбца любого отношения существует одно и только одно значение;

все операции выполняются над целым отношением, и результатом их выполнения также является целое отношение. Этот принцип называется **замыканием**.

# Физическая модель

---

**Физическая модель** - отражает все свойства (атрибуты) информационных объектов базы и связи между ними с учетом способа их хранения - используемой СУБД.

**Реляционная БД** на физическом уровне состоит из таблиц, между которыми могут существовать связи по ключевым значениям.

**Внутренняя модель** - база данных, соответствующая определенной физической модели.

**Внешняя модель** - комплекс программных и аппаратных средств для работы с базой данных, обеспечивающий процессы создания, хранения, редактирования, удаления и поиска информации, а также решающий задачи выполнения необходимых расчетов и создания выходных печатных форм.



**При создании проекта информационной системы для проектирования ее базы данных следует определить:**

---

*объекты информационной системы (сущности в концептуальной модели);*

*их свойства (атрибуты);*

*взаимодействие объектов (связи) и информационные потоки внутри и между ними.*

# Сущность

---

**Сущность** - некоторый обособленный объект или событие, информацию о котором необходимо сохранять в базе данных, имеющий определенный набор свойств - *атрибутов*.

Сущности могут быть:

физические (реально существующие объекты)

СТУДЕНТ, *атрибуты* - № зачетной книжки, фамилия, его факультет, специальность, № группы и т.д.,

абстрактные

ЭКЗАМЕН, *атрибуты* - дисциплина, дата, преподаватель, аудитория и пр.

Для *сущностей* различают ее тип и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр - конкретными значениями свойств.

# Атрибуты сущности

---

**Идентифицирующие и описательные.** Идентифицирующие *атрибуты* имеют уникальное значение для *сущностей* данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры *сущности*. Из потенциальных ключей выбирается один **первичный ключ** (ПК). В качестве ПК обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. ПК должен включать в свой состав минимально необходимое для идентификации количество *атрибутов*. Остальные *атрибуты* называются описательными.

**Простые и составные.** Простой *атрибут* состоит из одного компонента, его значение неделимо. Составной *атрибут* является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, адрес). Решение о том, использовать составной *атрибут* или разбивать его на компоненты, зависит от особенностей процессов его использования и может быть связано с обеспечением высокой скорости работы с большими базами данных.

**Однозначные и многозначные** - могут иметь соответственно одно или много значений для каждого экземпляра сущности.

**Основные и производные.** Значение основного *атрибута* не зависит от других *атрибутов*. Значение производного *атрибута* вычисляется на основе значений других *атрибутов* (например, возраст человека вычисляется на основе даты его рождения и текущей даты).

# Домен

---

**Домен** - это набор всех допустимых значений, которые может содержать *атрибут*. Понятие "**домен**" часто путают с понятием "тип данных". Необходимо различать эти два понятия. Тип данных - это физическая концепция, а домен - логическая.

Например, "целое число" - это тип данных, а "возраст" - это домен.

# СВЯЗИ

---

**Связи** - на концептуальном уровне представляют собой простые ассоциации между *сущностями*.

Существует несколько типов связей между двумя *сущностями*:

**«один к одному»,**

**«один ко многим»**

**«многие к одному»**

**«многие ко многим».**

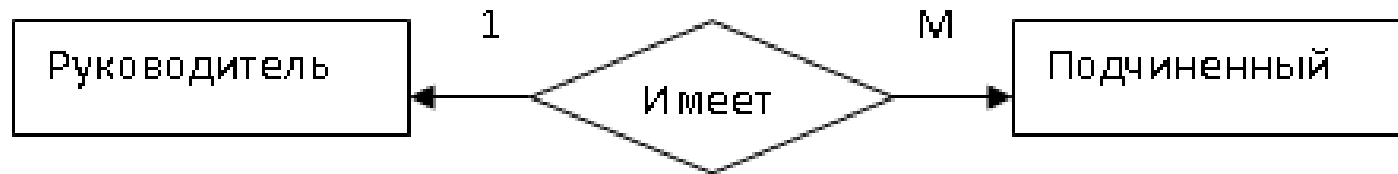
## Связь один к одному (1:1)

---

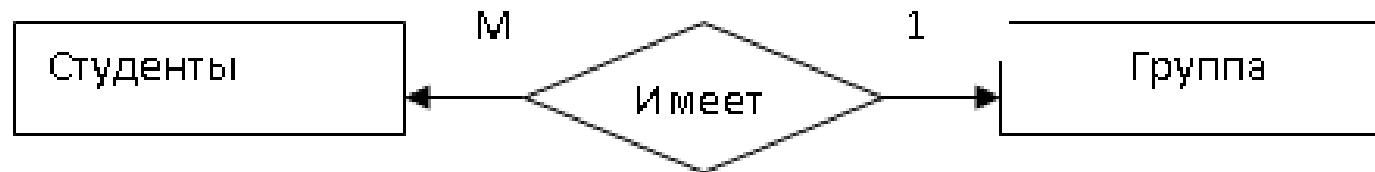


## Связь один к многим (1:M)

---

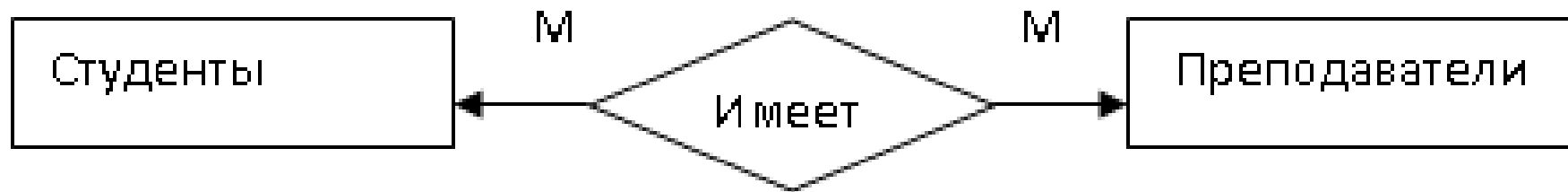


## Связь многие к одному (M:1)



## Связь многие ко многим (M:M)

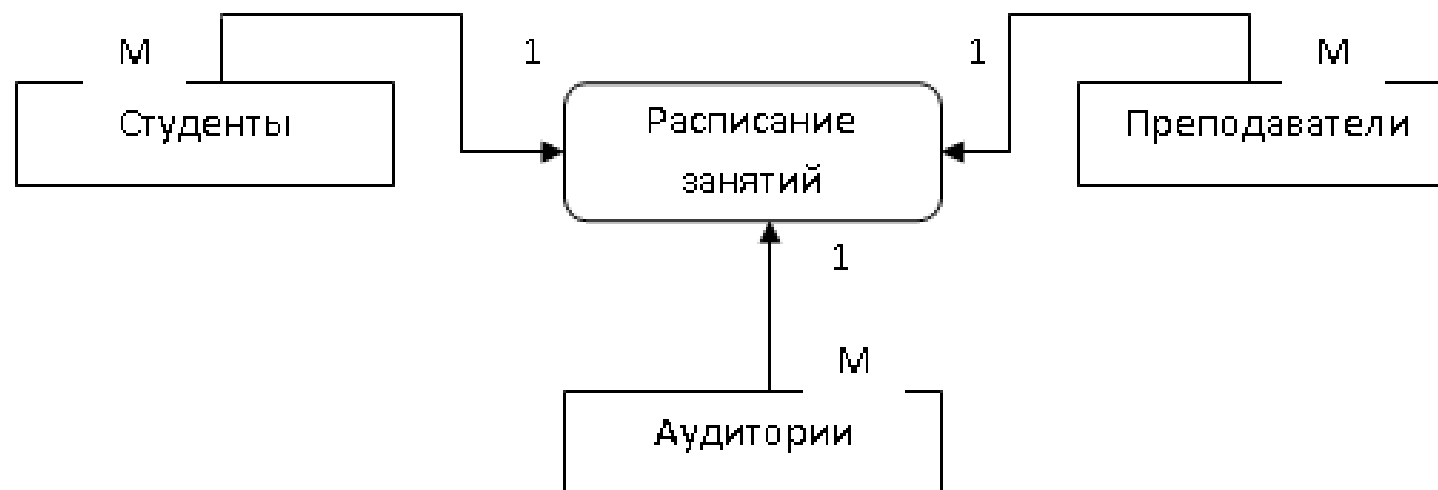
---





# Тренарные связи

---



# Основные этапы проектирования баз данных

---

**Системный анализ и словесное описание информационных объектов предметной области.** Подробное описание информации об объектах, формулировка задач с кратким описанием алгоритмов их решения, описанием входных и выходных документов.

**Проектирование инфологической модели предметной области** – частично формализованное описание объектов предметной области в терминах инфологической, например, ER-модель. Инфологическая модель отражает описание объектов, их свойства и взаимосвязи в виде схем.

**Даталогическое проектирование базы данных,** то есть описание базы данных в терминах принятой даталогической модели. На настоящий момент для этой цели используется реляционная модель данных.

**Физическое проектирование базы данных,** т.е. способ размещения базы данных на внешних носителях.

# Нормализация отношений

---

Процесс последовательного перехода к полным декомпозициям файлов БД называется нормализацией файлов БД, главная цель которой – исключение дублирования информации и потери присоединенных записей.

# Первая нормальная форма (1НФ)

Отношение называется нормализованным или приведенным к 1НФ, если все его атрибуты простые (далее неделимы). Преобразование отношения к 1НФ может привести к увеличению количества реквизитов (полей) таблицы и изменению ключа.

## *ПРИМЕР*

*Сотрудник (Номер, Фамилия, Имя, Отчество, Дата рождения, Отдел)*

~~*Сотрудник1 (Номер, Фамилия, Имя, Отчество, Отдел, Дети)*~~

*Сотрудник1 (Номер, Фамилия, Имя, Отчество, Отдел);*

*Дети (Номер, Имя ребенка, Возраст)*

# Вторая нормальная форма (2НФ)

**Файл находится во 2НФ, если он находится в 1НФ и все его поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом.**

## ПРИМЕР

**1НФ**

Студент (Номер, Фамилия, Имя, Отчество, Группа)

**2НФ**

**1НФ**

Успеваемость (Номер, Фамилия, Имя, Отчество, Дисциплина, оценка)

**2НФ**

Дисциплина (Дисциплина, Оценка)

Успеваемость (Номер, Фамилия, Имя, Отчество, Дисциплина)

# Нормализация отношений

## Третья нормальная форма (3НФ)

*Файл находится в 3НФ, если он находится во 2НФ и ни одно из его не ключевых полей не зависит функционально от любого другого не ключевого поля.*

«Расщепление» информационного объекта



# Нормальная форма Бойса-Кодда

---

**Нормальная форма Бойса-Кодда (англ. Boyce-Codd normal form; сокращённо BCNF) — одна из возможных нормальных форм отношения в реляционной модели данных.  
(усиленная третья нормальная форма)**

Переменная отношения находится в нормальной форме Бойса-Кодда, тогда и только тогда, когда **детерминанты всех ее функциональных зависимостей являются потенциальными ключами!**

## **Историческая справка**

Названа в честь Рэя Бойса и Эдгара Кодда, хотя Кристофер Дейт указывает, что на самом деле строгое определение <третьей> нормальной формы, эквивалентное определению нормальной формы Бойса-Кодда, впервые было дано Иэном Хитом (англ. Ian Heath) в 1971 году, поэтому данную форму следовало бы называть "нормальной формой Хита".

# Первый пример

Исходная таблица:

---

Номер клиента	Дата собеседования	Время собеседования	Номер комнаты	Номер сотрудника
C345	13.10.03	13.00	103	A138
C355	13.10.03	13.05	103	A136
C368	13.09.03	13.00	102	A154
C366	13.09.03	13.30	105	A207

В результате приведения к форме Бойса-Кодда получаются две таблицы:



Номер клиента	Дата собеседования	Время собеседования	Номер сотрудника
C345	13.10.03	13.00	A138
C355	13.10.03	13.05	A136
C368	13.09.03	13.00	A154
C366	13.09.03	13.30	A207

Дата собеседования	Номер комнаты	Номер сотрудника
13.10.03	103	A138
13.10.03	103	A136
13.09.03	102	A154
13.09.03	105	A207

## «Расщепление» информационного объекта

---

**Студент группы**

Номер\*  
Фамилия  
Имя  
Отчество  
Дата  
Группа  
Староста

**Студент**

Номер\*  
Фамилия  
Имя  
Отчество  
Дата  
Группа

+

=

**Группа**

Группа\*  
Староста

# Типовая последовательность работ по построению инфологической модели

---

- 1. Выделение в предметной области сущностей.**
- 2. Введение множества атрибутов для каждой сущности и выделение из них ключевых.**
- 3. Исключение множества повторяющихся атрибутов (при необходимости).**
- 4. Формирование связей между сущностями.**
- 5. Исключение связей типа M:N (при необходимости).**

# ER-моделирование

---

ER-модель — это модель данных, позволяющая описывать концептуальные схемы предметной области. Основная задача, которую решает ER-модель единое и наиболее понятное представления данных, как для разработчиков, так и для конечных пользователей БД.

**Модель представляет собой подход к проектированию баз данных. Основная идея этого подхода, модель сущность – связь, которая позволяет представить в формализованном виде объекты реального мира.**

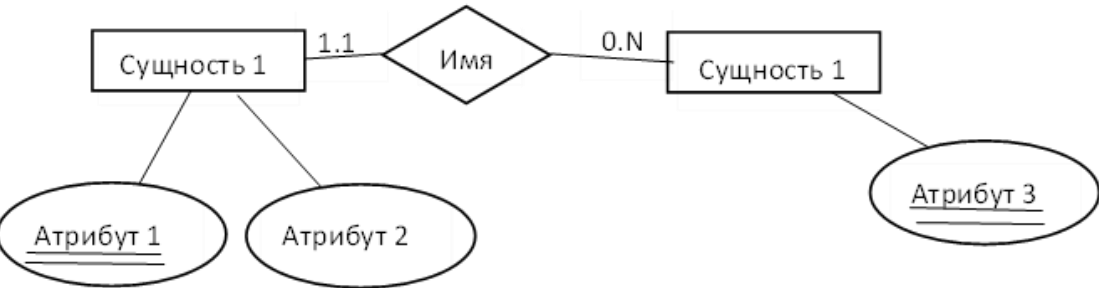
Каждый объект(сущность) представляет из себя набор некоторых отличительных свойств(атрибутов), которые в свою очередь могут иметь некоторые значения. Каждый объект может иметь отношения с другим объектом, что и отражает название модели — сущность – связь.

# НОТАЦИИ ER-МОДЕЛИРОВАНИЯ

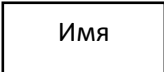

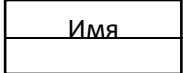
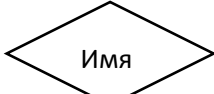
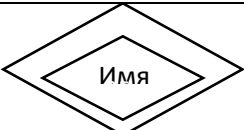

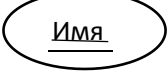


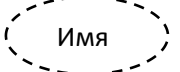
## Нотация Чена

Нотация Чена предоставляет большой набор средств моделирования данных включая, собственно, ERD, а также диаграммы атрибутов и диаграммы декомпозиции.

Эти диаграммные техники используются прежде всего для проектирования реляционных баз данных (хотя также могут с успехом применяться и для моделирования как иерархических, так и сетевых баз данных).



Пример схемы

Элемент диаграммы	Обозначение
	Независимая сущность
	Зависимая сущность
	Родительская сущность в иерархической связи
	Связь
	Идентифицирующая связь
	Атрибут
	Первичный ключ
	Внешний ключ
	Многозначный атрибут
	Получаемый атрибут в иерархических связях

Графическое обозначение элементов модели

# Питер Чэнь Пиньшань

## *Peter Pin-Shan Chen* 陳品山

---

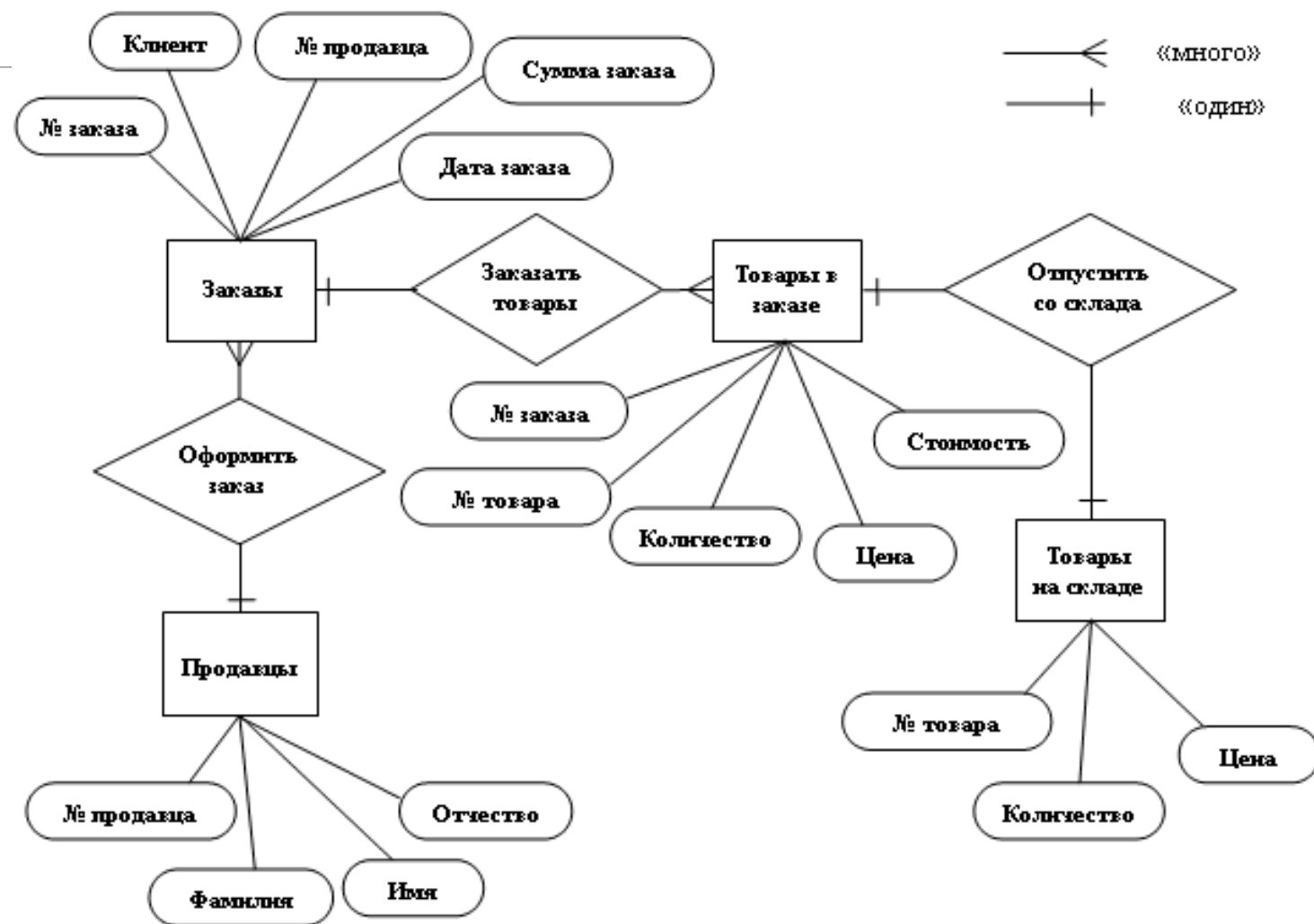
Впервые нотация была представлена Питером Ченом (американским учёным в области информатики) предложившим в 1976 году ER-модель данных.

ER-диграмма строится из элементов представленных в ER-модели, а именно: сущности, атрибуты, связи.



# Диаграмма «сущности-связи» ER-модель данных (Entity Relationship)

Peter Pin-Shan Chen



# Нотация Мартина (Crow`s Foot)

Данная нотация является одной из наиболее известных в разработке баз данных, отражающей уровень логического представления базы данных с обозначением некоторых компонентов модели базы данных в графическом виде, облегчая, тем самым, отображение диаграммы в рабочем пространстве.

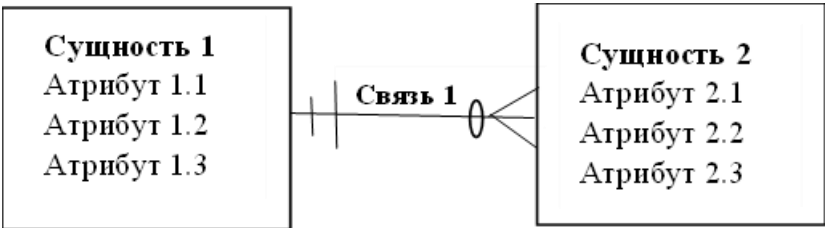
Основу всей модели базы данных в нотации Мартина составляют элементы "Сущность".

- Устанавливая связи между сущностями, в нотации Мартина её смысловое наполнение можно обозначать единственной глагольной формой, имеющей смысл связи от "левой" сущности к "правой" сущности, представляя в качестве "левой" сущности ту, у которой множественность связи в верхней её границе равна "1".

Обозначение	Кардинальность
—	нет
—	1,1
—○+	0,1
—<=	M,N
—○<=	0,N
— <=	1,N

Графическое обозначение элементов модели

Элемент диаграммы	Обозначение
<div>Имя</div>	Независимая сущность
<div>Имя</div>	Зависимая сущность
<div>Имя</div>	Родительская сущность в иерархической связи



Пример схемы



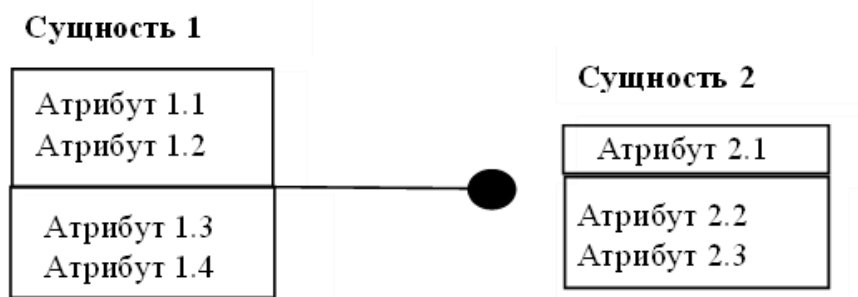
# Нотация IDEF1X

Методология моделирования IDEF1X, являясь расширением стандарта IDEF1, предназначена **для описания данных** (информации).

В её основе лежит язык семантического моделирования, основанного на концепции "сущность — связь", позволяющей определять данные и связи между ними. Методология используется для создания информационной модели предметной области с помощью идентификации ее сущностей и связей между ними.

Основными элементами модели IDEF1X являются сущности, атрибуты и отношения.

Атрибуты и отношения ничем не отличаются от тех, что представлены в модели Чена.



Пример схемы

Элемент диаграммы	Обозначает
	идентифицирующая связь
	неидентифицирующая связь

Графическое обозначение элементов модели

Элемент диаграммы	Обозначает
	1,1
	0,M
	0,1
	1,M
	точно N (N - произвольное число)

Обозначение элементов множественности

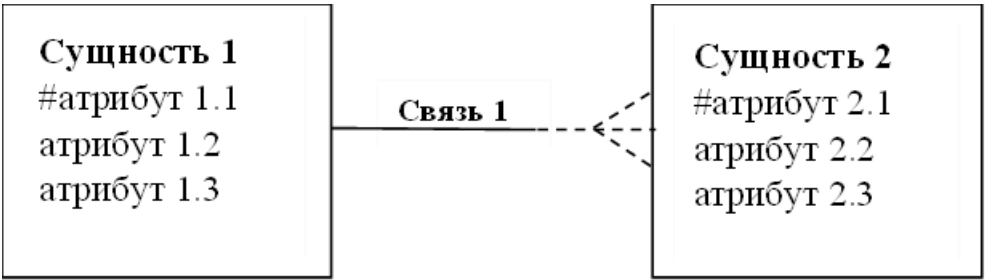
# Нотация Баркера

Все связи в нотации Баркера являются бинарными и представляют собой линии, соединяющие сущности. Для каждого конца связи должно быть определено имя и степень множественности, т.е. один или много объектов участвует в связи.

**Степень множественности** называется так же *кардинальным числом*.

Понятия категории и общей сущности в классической модели Чена соответствуют эквивалентным понятиям подтип и супертип в нотации Баркера.

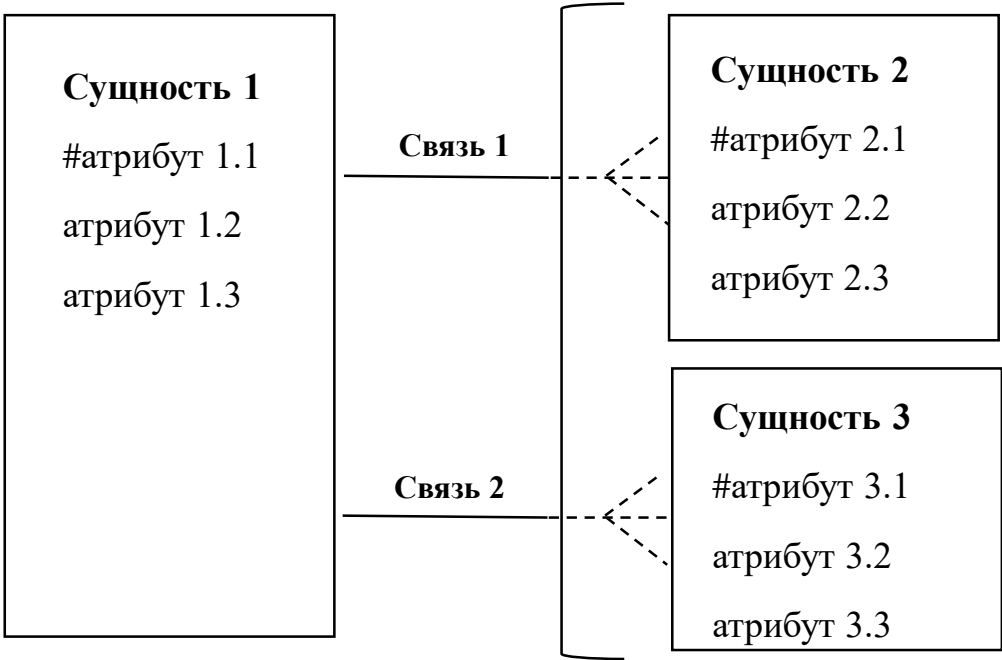
При существовании нескольких возможных ключей один из них обозначается в качестве первичного ключа, а остальные называются альтернативными ключами. В нотации Баркера модель данных может содержать сущности супертипы и подтипы, а также рекурсивные связи, связывающие сущность саму с собой .



Пример схемы

Графическое обозначение элементов модели

Обозначение	Кардинальность
-----	0,1
-----	1,1
----->	0,N
----->	1,N



Общая сущность

Сущность – категории

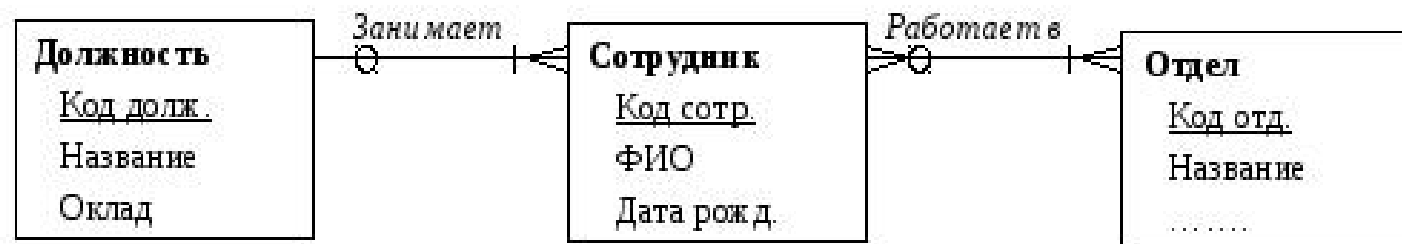
Для обозначения отношения категоризации вводится элемент "дуга"

# Сравнение нотаций

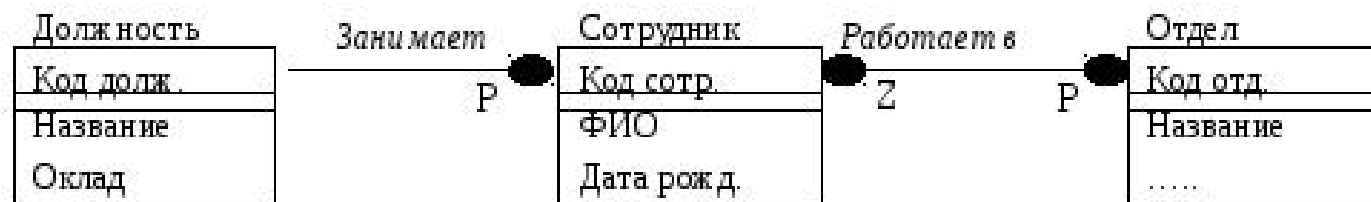
Нотация  
Чена



Нотация  
Мартина



Нотация  
IDEF1X



Нотация  
Баркера



# Описание модели данных информационной системы

## "Контингент студентов университета"

---

### **Постановка задачи.**

Главная задача системы - сохранение в базе данных всех необходимых сведений о студентах и их успеваемости, формирование необходимых печатных форм для проведения зачетной и экзаменационной работы преподавателей, генерация сводных итогов по результатам сессии для руководящих работников деканатов, институтов и университета. При разработке системы следует учитывать, что она взаимодействует с системами "Абитуриент", "Стипендия" и "Кадры университета". Информация о студентах первоначально поступает из системы "Абитуриент" и редактируется на уровне деканатов. Она должна также удовлетворять требованиям бухгалтерского учета по начислению стипендий. Система должна использовать справочник специальностей, утвержденный в вышестоящем министерстве. Информация об успеваемости студентов накапливается постоянно и сохраняется за весь период обучения, после чего переносится в архивное хранилище данных. В системе должен использоваться единый справочник дисциплин (предметов) для всех подразделений университета.

## Минимальный состав атрибутов и их описание для сущности "Студент"

Имя атрибута	Описание, особенности использования
Номер зачетки	Первичный ключ - уникальный номер, однозначно идентифицирующий студента университета
Фамилия, имя, отчество	Является простым с точки зрения экземпляра сущности, при необходимости из общего поля можно выделить составляющие его фамилию, имя и отчество или фамилию и инициалы, однако на практике часто этот атрибут разделяют на 3 отдельных; первый вариант является более экономичным по необходимой общей ширине поля таблицы
Дата поступления в университет	В нашей стране наиболее часто используется формат работы с датой в виде ДД.ММ.ГГ, что совпадает с немецким (German) форматом дат. Количество цифр года: либо две - для новых систем, поддерживающих заданный в Microsoft Windows годичный интервал (Панель управления - Язык и стандарты - Дата - "При вводе двух цифр года воспринимать их как год между:"), или для систем, в которых аналогичный интервал может быть задан в программе, - либо 4 цифры
Факультет (№ факультета)	Может быть сложным (кроме кода и названия, может содержать и другие сведения); даже в том случае, если для сущности "Студент" мы хотим сохранять название факультета, оно должно быть представлено в одинаковом виде для каждого факультета, поэтому, в соответствии с принципами нормализации баз данных, этот атрибут следует представить в виде номера, являющегося внешним ключом для новой сущности - "Факультет", в которой каждому номеру, являющемуся первичным ключом, будут соответствовать название и прочие атрибуты этой сущности
Специальность(код специальности)	Может быть сложным, кроме того, необходимо использовать справочник министерства с утвержденными кодами специальностей, поэтому данный атрибут должен хранить код специальности - внешний ключ для первичного ключа новой сущности "Специальность"
Курс	Число от 1 до 5
Номер группы	Трехзначное число
Номер паспорта	Состав и вид паспортных данных определяется требованиями бухгалтерской отчетности перед налоговыми органами, фондами социального страхования и пенсионным фондом
...	Прочие атрибуты, которых может быть достаточно много

### Атрибуты сущности «Успеваемость»

Имя атрибута	Описание, особенности использования
Номер зачетки	Внешний ключ (к сущности "Студент")
Номер семестра	Число от 1 до 10
Предмет (№ предмета)	Может быть сложным, его следует заменить на его номер (внешний ключ) и связать с новой сущностью "Предмет", состоящий, как минимум, из атрибутов "номер предмета" (первичный ключ) и "название предмета"
Оценка	Может быть представлена цифрами от 0 до 5 или 1 буквой: например "н" - неявка
Дата получения оценки	Формат даты обычно ДД.ММ.ГГ
Фамилия преподавателя	Это поле может быть связано с сущностью "Преподаватель". В данном учебном примере ограничимся простым атрибутом
...	Могут быть добавлены и другие атрибуты, например, номер экзаменационной ведомости

## Атрибуты сущности "Факультет"

Имя атрибута	Описание, особенности использования
Номер факультета	Первичный ключ
Название факультета	Может быть достаточно длинным, но не более 255 символов
...	Могут быть добавлены и другие атрибуты, например, декан, номер комнаты деканата и т.д.

## Атрибуты сущности "Специальность"

Имя атрибута	Описание, особенности использования
Код специальности	Первичный ключ - значение из справочника министерства
Название специальности	Значение из справочника министерства
...	Могут быть добавлены и другие атрибуты



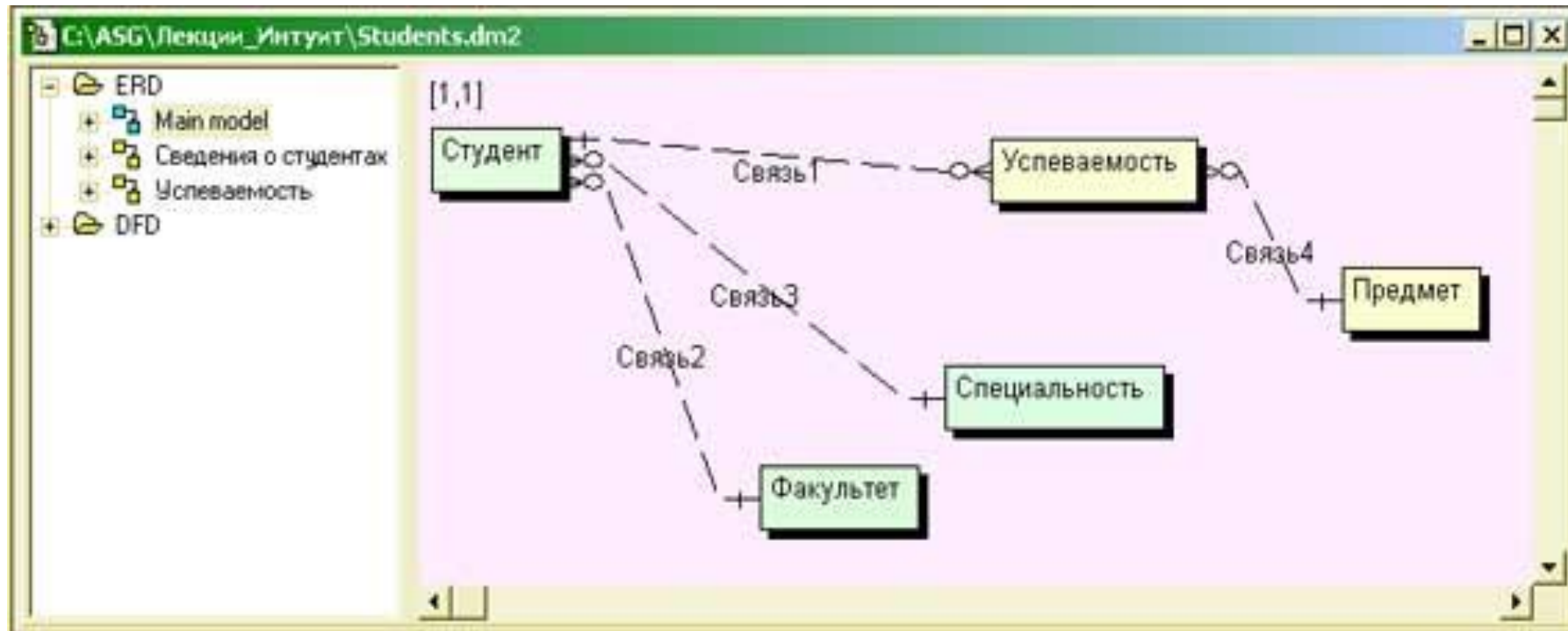
## Атрибуты сущности "Предмет"

Имя атрибута	Описание, особенности использования
№ предмета	Первичный ключ
Название предмета	Общий справочник университета
...	Могут быть добавлены и другие атрибуты

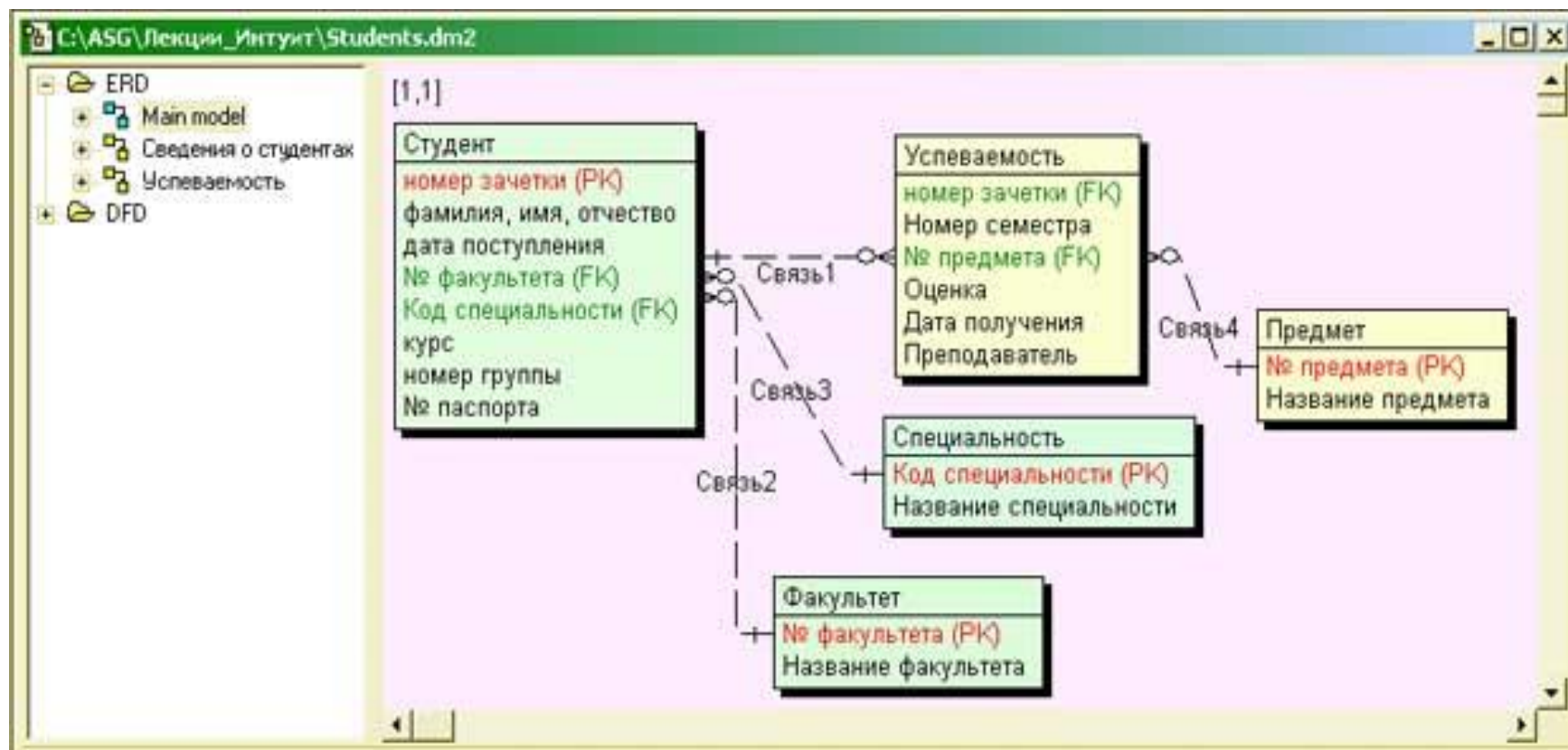
## Состав базы данных информационной системы

№ п/п	Сущности концептуальной модели	Таблицы физической модели	
		Названи е	Информация
1.	"Студент"	"SPISOK"	"Список студентов"
2.	"Успеваемость"	"OCENKI "	"Оценки студентов"
3.	"Факультет"	"FCLT"	Справочник факультетов
4.	"Специальность"	"SPECT"	Справочник специальностей
5.	"Предмет"	"PREDME T"	Справочник предметов

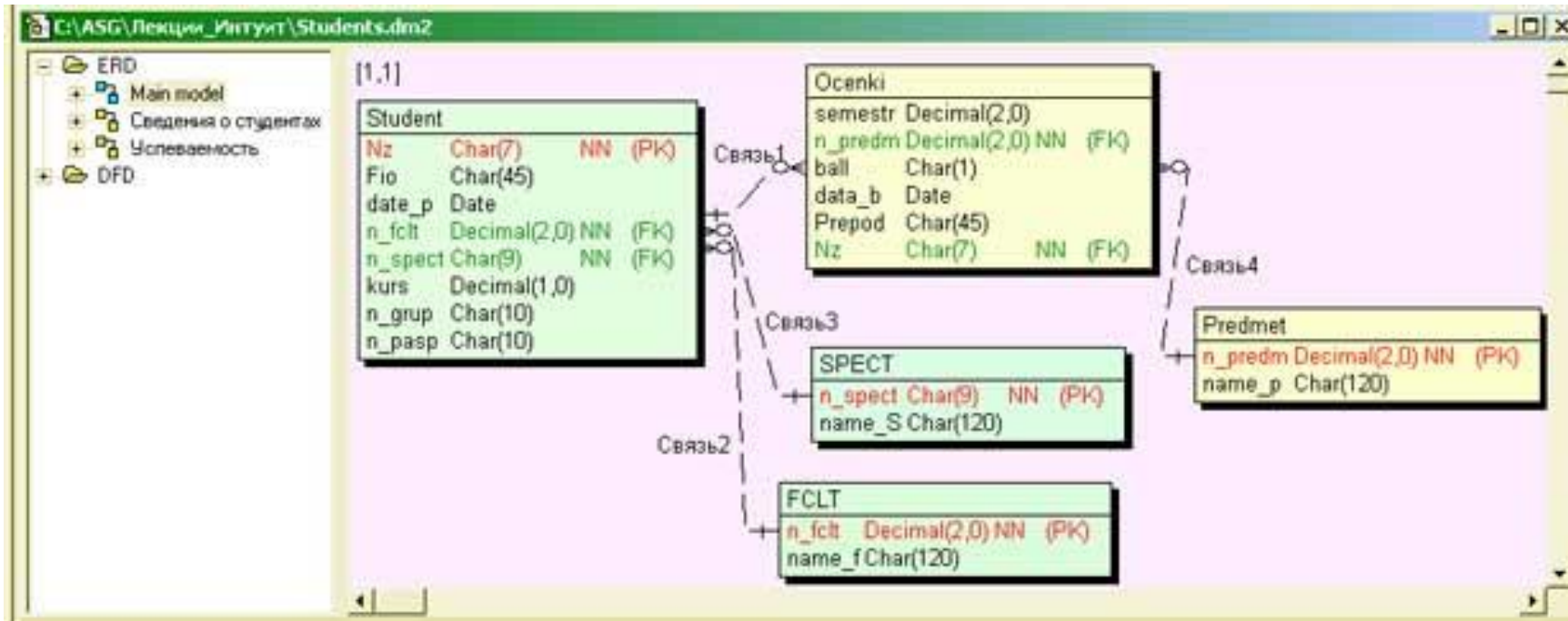
## Простейший вид ER-диаграммы в системе CASE Studio



# Концептуальная модель базы данных



# Физическая модель базы данных



## Статистические функции

---

Функция	Описание
Sum	<i>сумма значений некоторого поля для группы</i>
Avg	<i>среднее от всех значений поля в группе</i>
Max	<i>максимальное значение в группе</i>
Min	<i>минимальное значение в группе</i>
Count	<i>число значений поля в группе без учета пустых значений</i>
Stdev	<i>среднеквадратичное отклонение от среднего значения в поле</i>
Var	<i>дисперсия значений поля в группе</i>
First Last	<i>значения поля из первой или последней записи в группе</i>

# Источники

---

1. [UML\_class] Васильев В.С. Диаграммы классов UML. URL: [https://pro-prof.com/archives/3212#page\\_2\\_1](https://pro-prof.com/archives/3212#page_2_1)
2. [Chen\_model] Васильев В.С. Пример проектирования простой базы данных в MS SQL. URL: <https://pro-prof.com/archives/7750>
3. [ER\_ivanova] Иванова Е.В. Модель «сущность-связь» (Технологии баз данных. Лекция 3). URL: <https://foreva.susu.ru/courses/db/lecture3.pdf>
4. [Plantuml] PlantUML Web Server. URL: <http://www.plantuml.com>
5. [yEd] yEd graph editor. URL: <https://www.yworks.com/products/yed>
6. [ER\_Kravishein] Кривошеин М. ER: диаграммы сущность – связь. URL: [http://mikkri.narod.ru/library/pdf/ER\\_Modeling.pdf](http://mikkri.narod.ru/library/pdf/ER_Modeling.pdf)
7. [UML\_ISO\_1] ISO/IEC 19505-1:2012 «Информационные технологии. Унифицированный язык моделирования группы по управлению объектами (OMG UML). Часть 1. Инфраструктура» (Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 1: Infrastructure).
8. [UML\_ISO\_2] ISO/IEC 19505-2:2012 «Информационные технологии. Унифицированный язык моделирования группы по управлению объектами (OMG UML). Часть 2. Сверхструктура» (Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 2: Superstructure).
9. [IDEF1X\_ISO] ISO/IEC/IEEE 31320-2:2012. Information technology — Modeling Languages — Part 2: Syntax and Semantics for IDEF1X97 (IDEFobject). Информационные технологии. Языки моделирования. Часть 2. Синтаксис и семантика для IDEF1X97 (IDEFobject). 320 стр.
10. [Anisimov\_IDEF1X] Анисимов В.В. Концептуальное проектирование с использованием методологии IDEF1X. URL: [https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema7/tema7\\_2](https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema7/tema7_2)