

# Лекция 4

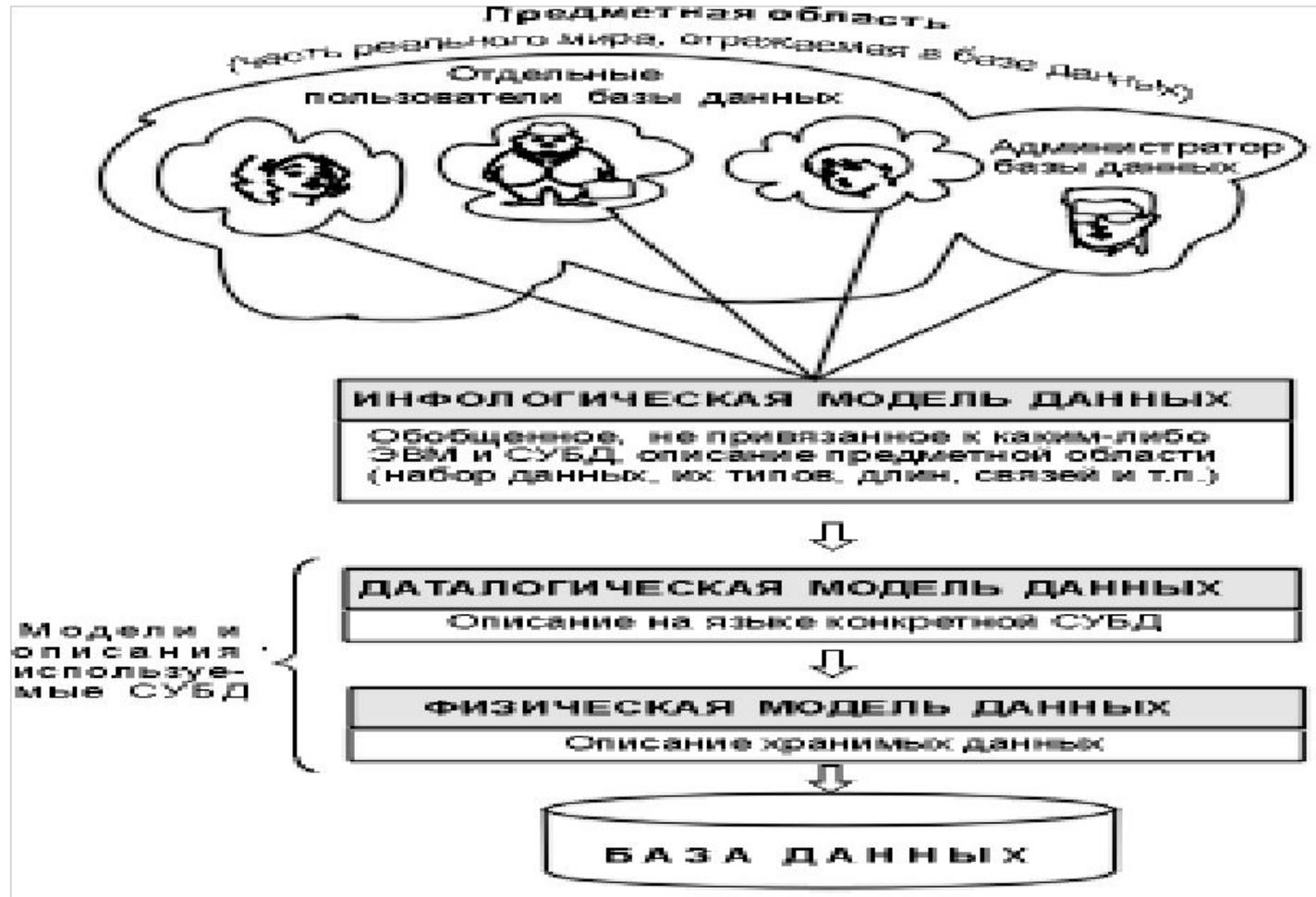
## Логическое проектирование БД

- Цель и способы логического проектирования БД
- Правила порождения реляционных отношений из инфологической модели “сущность-связь”
- Избыточное дублирование данных и аномалии.
- Понятие нормализации и метод нормальных форм. Зависимости между атрибутами

Используют теоретические языки запросов, наиболее распространенным из которых является SQL.

Классической технологией проектирования реляционных БД является применение теории НОРМАЛИЗАЦИИ отношений – МЕТОД НОРМАЛЬНЫХ ФОРМ. Этот этап часто называют этапом построения ДАТАЛОГИЧЕСКОЙ (логической) модели.

Последним этапом является ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БД – выбор рациональной структуры данных и методов доступа к ним (имея в виду особенности конкретной СУБД – например, ACCESS).



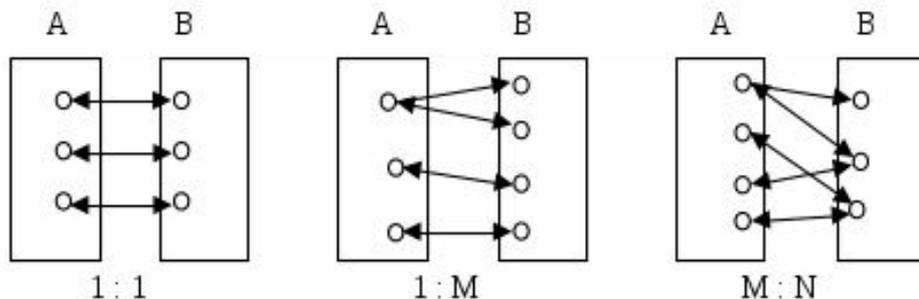
Этапы проектирования БД

- На следующем этапе проектирования строится логическая (дatalogическая) модель базы данных, представляющая собой описание сущностей и связей ER-модели в терминах реляционной модели данных.
- Преобразование ER-модели в реляционную может быть произведено с использованием так называемых CASE-средств (средств автоматизированного проектирования баз данных), в частности, пакета ERWin .

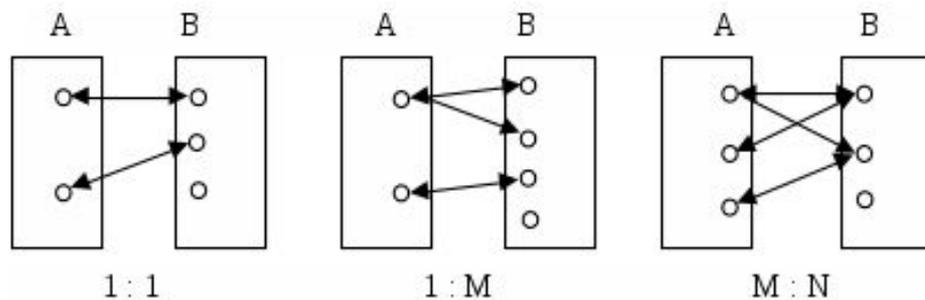


## Классификация связей по классу принадлежности:

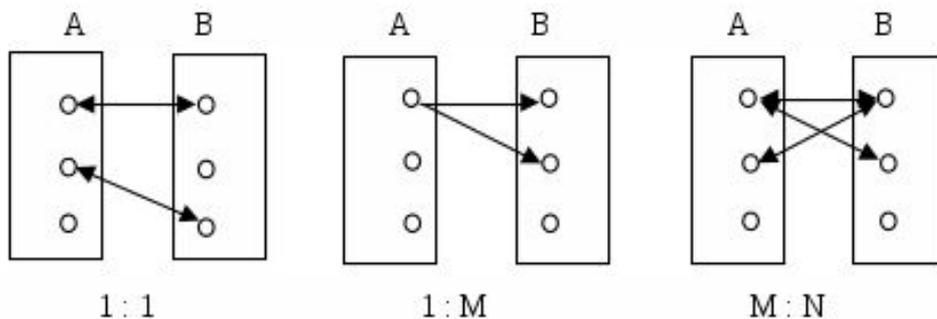
1) обязательный класс принадлежности обоих типов сущностей



2) класс принадлежности сущности типа А обязательный, В необязательный



3) класс принадлежности обоих типов сущностей необязательный



# Правила порождения реляционных отношений из инфологической модели “сущность-связь”.

- 
- Порождение реляционных отношений и связей между ними из ER-модели можно разделить на 2 этапа: преобразование сущностей и преобразование связей.

# Правила преобразования сущностей

- 1. Каждой сущности ставится в соответствие отношение реляционной модели.**

При этом имена сущности и отношения могут быть различными, потому что на имена сущностей могут не накладываться дополнительные синтаксические ограничения, кроме уникальности имени в рамках модели. Имена отношений могут быть ограничены требованиями конкретной СУБД, чаще всего эти имена являются идентификаторами в некотором базовом языке, они ограничены по длине и не должны содержать пробелов и некоторых специальных символов. Например, сущность может быть названа "Книжный каталог", а соответствующее ей отношение желательно назвать, например, BOOKS (без пробелов и латинскими буквами).

**2. Каждый атрибут сущности становится атрибутом отношения, которому приписывают тип данных и свойство обязательности или необязательности данного атрибута (допустимости или недопустимости неопределенного значения).**

Переименование атрибутов должно происходить в соответствии с теми же правилами, что и переименование отношений в п.1. Для каждого атрибута задается конкретный допустимый в СУБД тип данных и обязательность или необязательность данного атрибута (то есть допустимость или недопустимость NULL значений для него).

**3. Первичный ключ сущности становится первичным ключом отношения. Атрибуты, входящие в первичный ключ, получают свойство обязательности (NOT NULL).**

**4. В каждое отношение, соответствующее дочерней сущности, добавляется набор атрибутов родительской сущности, являющийся первичным ключом родительской сущности. В отношении, соответствующем дочерней сущности, этот набор атрибутов становится внешним ключом (FOREIGN KEY ).**

**5. Для установки необязательного класса принадлежности у атрибутов, соответствующих внешнему ключу, устанавливается свойство допустимости неопределенных значений (признак NULL) . При обязательном классе принадлежности эти атрибуты получают свойство недопустимости неопределенных значений (признак NOT NULL) .**

# Правила преобразования сущностей

- Сущность  Отношение
- Атрибут сущности  Атрибут отношения (+ тип данных, + ОКП/НКП – обязательный/необязательный класс принадлежности)
- Первичный ключ сущности  Первичный ключ отношения (ОКП - Not Null)
- Дочерняя сущность  Дочернее отношение (+ атрибуты первичного ключа родительской сущности (FK - FOREIGN KEY ))
- Устанавливаются классы принадлежности и свойства допустимости /недопустимости неопределенных значений у атрибутов, соответствующих внешнему ключу (если НКП – NULL, если ОКП – NOT NULL).

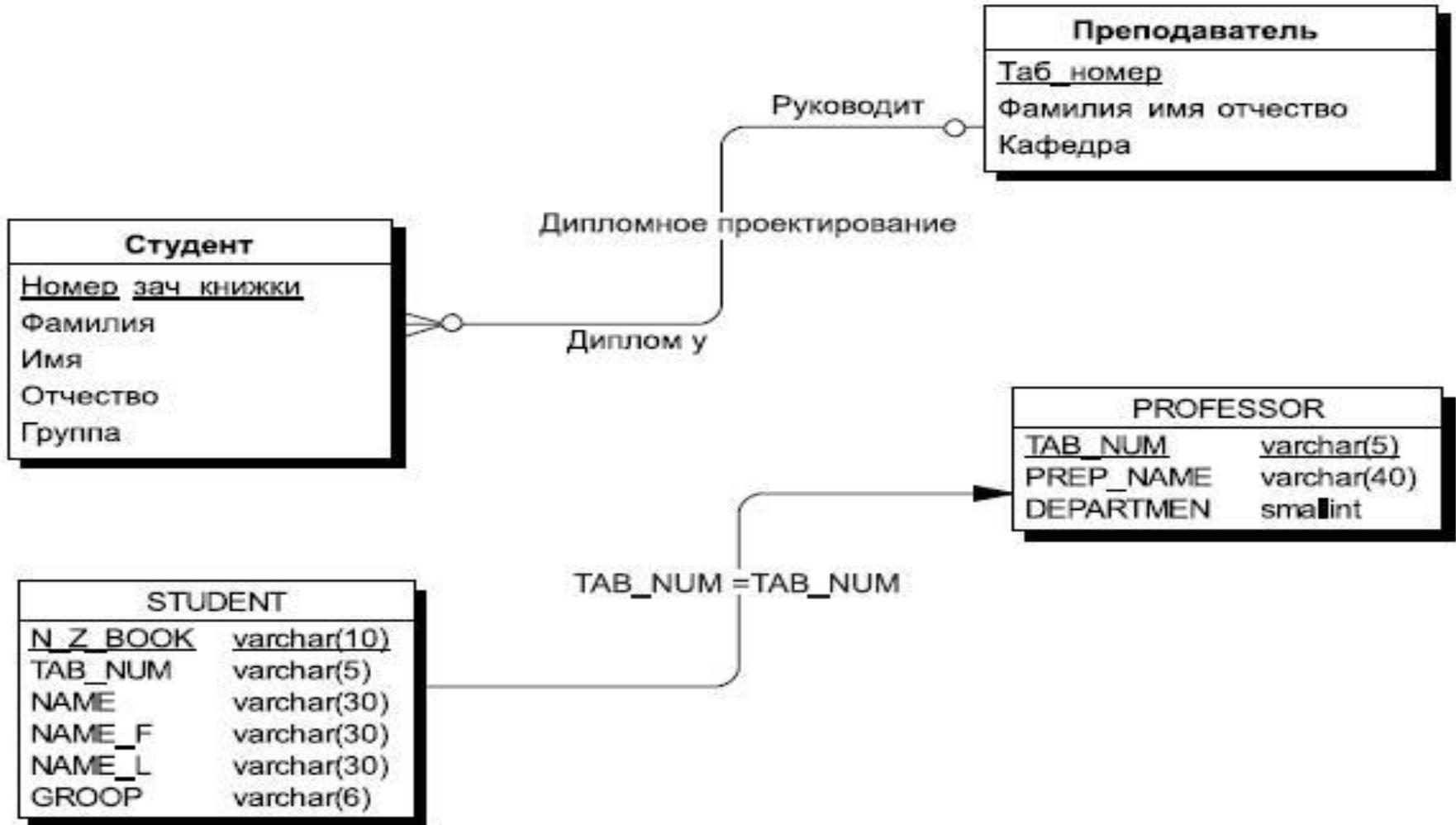
СОТРУДНИК
<u>Табельный номер</u>
Фамилия
Имя
Отчество
Количество детей

EMPLOYEE	
<u>T_NUM</u>	int
NAME	varchar(30)
F_NAME	varchar(30)
L_NAME	varchar(30)
COUNT_CH	varchar(30)

Свойства атрибутов отношения EMPLOYEE

Column	Code	Type	
COUNT	CH	Varchar(30)	Not Null
F	NAME	Varchar(30)	Not Null
L	NAME	Varchar(30)	Null
NAME		Varchar(30)	Null
T	NUM	Int	Not Null

# Преобразование взаимосвязанных **сущностей** СТУДЕНТ и ПРЕПОДАВАТЕЛЬ к взаимосвязанным отношениям **STUDENT** и **PROFESSOR**



# Правила преобразования связей

## Формализация бинарных связей

На основе анализа диаграмм ER-типа формируются отношения БД. При этом следует учитывать степень связи сущностей (1:1, 1:M, M:M) и класс принадлежности (КП) сущностей. Класс принадлежности сущностей - это степень участия сущности в связи.

Класс принадлежности сущности называется обязательным, если все экземпляры сущности участвуют в рассматриваемой связи **ОБЯЗАТЕЛЬНО**, в противном случае класс принадлежности НЕОБЯЗАТЕЛЬНЫЙ.

В зависимости от комбинаций КП и степени связи выделяют 6 ситуаций (правил), каждой из которых соответствует определенное проектное решение БД.

**1. Если степень бинарной связи 1:1 и КП обеих сущностей является обязательным, то требуется только одно отношение, ключом которого может быть ключ любой из двух сущностей. Пример:**

**Врач (Номер врача, Фамилия, Специальность)**

**Пациент (Регистрационный номер пациента, Номер койки, Фамилия пациента)**

**Если каждый врач лечит обязательно только одного пациента, а каждый пациент имеет обязательно только одного врача, то можно составить следующее отношение:**

**Лечение (Номер врача, Фамилия, Специальность, Регистрационный номер пациента, Номер койки, Фамилия пациента)**

**1:1**

**О - О**

<b>НП</b>	<b>ФИО</b>	<b>ТЕЛ</b>
<b>1</b>	<b>Иванов</b>	<b>212</b>
<b>2</b>	<b>Петров</b>	<b>213</b>
<b>3</b>	<b>Сидоров</b>	<b>214</b>

**И**

<b>НК</b>	<b>НазвКурса</b>
<b>100</b>	<b>Физика</b>
<b>200</b>	<b>Информатика</b>
<b>300</b>	<b>Философия</b>

**Результат**

<b>НП</b>	<b>ФИО</b>	<b>ТЕЛ</b>	<b>НК</b>	<b>НазвКурса</b>
<b>1</b>	<b>Иванов</b>	<b>212</b>	<b>100</b>	<b>Физика</b>
<b>2</b>	<b>Петров</b>	<b>213</b>	<b>200</b>	<b>Информатика</b>
<b>3</b>	<b>Сидоров</b>	<b>214</b>	<b>300</b>	<b>Философия</b>

**2. Если степень бинарной связи 1:1 и КП одной сущности является обязательным, а другой необязательным, то необходимо построение двух отношений. При этом к отношению, сущность которого имеет обязательный КП, добавляется ключ сущности с необязательным КП в качестве внешнего ключа. Пример:**

**Если каждый врач лечит обязательно только одного пациента, но имеются пациенты, которые не имеют (в силу каких-то причин) врача, то можно составить отношения:**

**Врач (Номер врача, Фамилия, Специальность, Регистрационный номер пациента (ФК) )**

**Пациент (Регистрационный номер пациента, Номер койки, Фамилия пациента)**

- ОДИН КУРС - ОДИН ПРЕПОДАВАТЕЛЬ, НО
- НЕТ ПРЕПОДАВАТЕЛЕЙ БЕЗ КУРСОВ (Обязательный КП):
- В отношении с обязательным КП добавляется ключ сущности с необязательным КП как FK:
- ПРЕПОДАВАТЕЛЬ(НП, НК (FK), ФИО, ТЕЛ)
- КУРС(НК, НазвКурс)

1:1

Н - О

НК (РК)

НазвКурса

100

Физика

200

Информатика

300

Философия

и

НП (РК)	НК (FK)	ФИО	Тел
1	100	Иванов	212
2	200	Петров	213
3	200	Сидоров	214

**3. Если степень бинарной связи 1:1 и КП обеих сущностей является необязательным, то необходимо построение трех отношений, одно из которых является сущностей-связкой двух исходных отношений. Эта сущность будет иметь по одному ключу от каждой сущности. Пример:**

**Если каждый врач лечит только одного пациента, а каждый пациент имеет только одного врача и при этом могут быть врачи без пациентов и пациенты без врачей, то можно составить следующие отношения:**

**Врач (Номер врача, Фамилия, Специальность)**

**Лечение (Номер врача, Регистрационный номер пациента)**

**Пациент (Регистрационный номер пациента,  
Номер койки, Фамилия пациента)**

Если каждый преподаватель читает только один курс и каждый курс читается только одним преподавателем, но имеются курсы без преподавателей и преподаватели без курсов:

ПРЕПОДАВАТЕЛЬ(НП, ФИО, ТЕЛ)  
КУРС(НК, НазвКурс)



Образуется новая сущность:

ПРЕП\_КУРС(НП, НК)

1:1

Н - Н

Курс

НК (PK)	НазвКурса
100	Физика
200	Информатика
300	Философия

Преподаватели

НП (PK)	ФИО	Тел
1	Иванов	212
2	Петров	213
3	Сидоров	214

НП+НК

НП	НК
1	100
3	300

Далее рассмотрим формирование отношений для связи 1:М. Для нашего примера это означает, что один врач может иметь одного или более пациентов. Отметим, что в этом случае проектное решение БД зависит от КЦ на стороне МНОГО.

4. Если степень бинарной связи 1:М и КЦ сущности на стороне МНОГО (М-связной сущности) является обязательным, то ключ отношения сущности на стороне **ОДИН** (односвязной сущности) добавляется как атрибут в отношение, соответствующее М-связной сущности. Пример:

Если в нашем примере нет пациентов без врачей, то решение может быть следующим:

**Врач (Номер врача, Фамилия, Специальность)**

**Пациент (Регистрационный номер пациента,  
Номер врача (FK), Номер койки, Фамилия  
пациента)**

## 1 : M (O)

**В случае объединения двух отношений в одну таблицу имеем аномалии:**

<b>НП</b>	<b>ФИО</b>	<b>Тел</b>	<b>НК</b>	<b>НазвКурс</b>
<b>1</b>	<b>Иванов</b>	<b>212</b>	<b>200</b>	<b>Информатика</b>
<b>2</b>	<b>Петров</b>	<b>213</b>	<b>300</b>	<b>Философия</b>
<b>1</b>	<b>Иванов</b>	<b>212</b>	<b>100</b>	<b>Физика</b>
<b>3</b>	<b>Сидоров</b>	<b>214</b>		

- ОДИН ПРЕПОДАВАТЕЛЬ ЧИТАЕТ МНОГО КУРСОВ, НО НЕ МОГУТ БЫТЬ КУРСЫ БЕЗ ПРЕПОДАВАТЕЛЕЙ:
- ЕСЛИ НЕ МОЖЕТ БЫТЬ КУРСОВ БЕЗ ПРЕПОДАВАТЕЛЕЙ - на стороне МНОГО (КУРС) ОБЯЗАТЕЛЬНЫЙ КП:
- Ключ отношения ПРЕПОДАВАТЕЛЬ добавляется в отношении КУРС:
- ПРЕПОДАВАТЕЛЬ(НП,ФИО,ТЕЛ)
- КУРС(НК, **НП (FK)**,НазвКурс)

**1 : M (O)**

**Препод.**

**1 : M**

**Курс**

<b>НП (PK)</b>
<b>ФИО</b>
<b>Тел</b>



<b>НК (PK)</b>
<b>НП (FK)</b>
<b>НазвКурс</b>

**5. Если степень бинарной связи 1:М и КП сущности на стороне МНОГО (М-связной сущности) является необязательным, то необходима сущность-связка, причем ключ каждого отношения служит ключом соответствующего отношения и одного отношения для связи. Отношение связка имеет в качестве ключа ключ из отношения на стороне МНОГО и ключ из отношения на стороне ОДИН. Пример:**

**Если в нашем примере могут быть пациенты без врачей, то решение может быть следующим:**

**Врач (Номер\_врача, Фамилия,  
Специальность)**

**Лечение (Номер\_врача ,**

**Регистрационный номер пациента),**

**Пациент (Регистрационный номер пациента,  
Номер койки, Фамилия пациента)**

**1 : M (H)**

**В случае объединения двух отношений в одну таблицу имеем аномалии:**

<b>НП</b>	<b>ФИО</b>	<b>Тел</b>	<b>НК</b>	<b>НазвКурс</b>
<b>1</b>	<b>Иванов</b>	<b>212</b>	<b>200</b>	<b>Информатика</b>
<b>2</b>	<b>Петров</b>	<b>213</b>	<b>300</b>	<b>Философия</b>
<b>1</b>	<b>Иванов</b>	<b>212</b>	<b>100</b>	<b>Физика</b>
			<b>500</b>	<b>Выч. Матем.</b>

- ОДИН ПРЕПОДАВАТЕЛЬ – МНОГО КУРСОВ
- ЕСЛИ МОЖЕТ БЫТЬ КУРС БЕЗ ПРЕПОДАВАТЕЛЯ (Т.Е. НЕОБЯЗАТЕЛЬНЫЙ КП), ТО ОБРАЗУЕТСЯ СУЩНОСТЬ - СВЯЗКА:
- ПРЕПОДАВАТЕЛЬ(НП,ФИО,ТЕЛ)
- ПРЕП\_КУРС(НП,НК)
- КУРС(НК, НазвКурс)

# Курс

НК (PK)	НазвКурса
100	Физика
200	Информатика
300	Философия

M:1

# Преподаватели

НП (PK)	ФИО	Тел
1	Иванов	212
2	Петров	213
3	Сидоров	214

НП+НК

НП	НК
1	200
3	100
2	200

**6. Если степень бинарной связи M:M (M:N), то (как и в правиле номер 5), необходима сущность связка. Заметим, что в этом случае КП даже не рассматривается.**

**Пример:**

**Если один врач может иметь много пациентов, и каждый пациент может лечиться у многих врачей, то решение может быть такое же, как и в случае 5:**

**Врач (Номер врача, Фамилия, Специальность)**

**Лечение (Номер врача,**

**Регистрационный номер пациента)**

**Пациент (Регистрационный номер пациента, Номер койки, Фамилия пациента)**

**Часто составной ключ в сущности-связке заменяется новым атрибутом-ключом:**

**Лечение (Номер записи в журнале, Номер врача, Регистрационный номер пациента)**

- ОДИН ПРЕПОДАВАТЕЛЬ – МНОГО КУРСОВ и ОДИН КУРС – МНОГО ПРЕПОДАВАТЕЛЕЙ:
- КП не исследуется, образуется сущность – связка:
- ПРЕПОДАВАТЕЛЬ(НП,ФИО,ТЕЛ)
- ПРЕП\_КУРС(НП,НК)
- КУРС(НК, НазвКурс)

Курс

**M:N**

Преподаватели

НК (PK)	НазвКурса
100	Физика
200	Информатика
300	Философия

НП (PK)	ФИО	Тел
---------	-----	-----

1	Иванов	212
---	--------	-----

2	Петров	213
---	--------	-----

3	Сидоров	214
---	---------	-----

НП+НК

НП	НК
1	200
1	100
2	100

Пример преобразования инфологической модели  
в реляционную.

Используем предметную область  
«Успеваемость студентов» из  
лабораторного практикума.  
(Построим инфологическую модель  
«задним числом»)

**Сущности предметной области(на языке  
инфологического моделирования ЯИМ)**

**Факультет (Код ф-та, Наименование,  
Декан)**

**Группа (Код группы, Номер, Год  
создания)**

**Студент (Код студента, № зачетки, ФИО,  
Балл ЕГЭ, Дата рождения, Город)**

**Дисциплина (Код дисц-ны. Название.**

**Связи между сущностями  
(на языке инфологического  
моделирования ЯИМ)**

**Факультет-группа [Факультет 1, Группа M]**

**(Код ф-та, Код группы)**

**Группа-студент [Группа 1, Студент M]**

**(Код группы, Код студента)**

**Студент-дисциплина [Студент M, Дисциплина  
N]**

**(Код студента, Код дисц-ны)**

Связь **Факультет-группа** является связью 1: М, так как на одном факультете может учиться много групп, но каждая группа относится только к одному факультету. Класс принадлежности сущности **Группа** – обязательный, так как каждая группа должна относиться к какому-то факультету. Класс принадлежности сущности **Факультет** – необязательный, так как факультеты могут существовать независимо от наличия групп в них.

Связь **Группа-студент** является связью 1: М, так как в одной группе может учиться много студентов, но каждый студент учится только в одной группе. Класс принадлежности сущности **Студент** – обязательный, так как каждый студент должен относиться к какой-то группе. Класс принадлежности сущности **Группа** – необязательный, так как группы могут существовать независимо от наличия студентов в них.

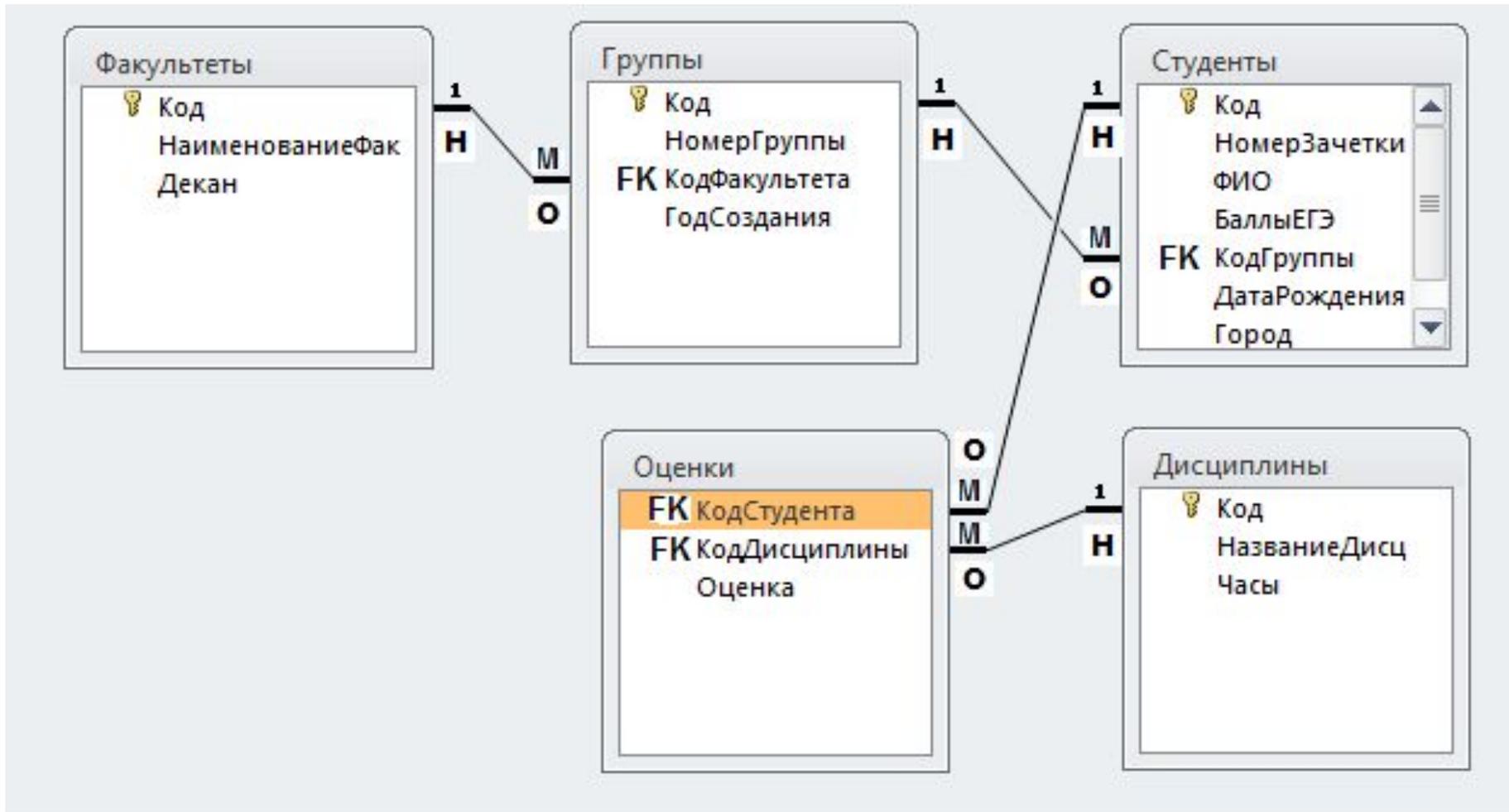
**Связь Студент-дисциплина** является связью М:М, так как каждый студент изучает много дисциплин, и каждая дисциплина изучается многими студентами. Класс принадлежности обеих сущностей – необязательный, так как не все студенты изучают некоторые дисциплины и не все дисциплины изучаются некоторыми студентами.

Независимой сущности **Факультет** поставим в соответствие отношение **Факультеты** с теми же атрибутами, что и у сущности. Первичный ключ сущности делаем первичным ключом отношения. Аналогичным образом преобразуем независимую сущность **Дисциплина** в отношение **Дисциплины**.

Сущность **Группа** преобразуем в отношении **Группы** с теми же атрибутами и первичным ключом. В соответствии с **правилом 4** преобразования связей добавим в это отношение атрибут **КодФакультета** в качестве внешнего ключа. Аналогичным образом преобразуем сущность **Студент** в отношении **Студенты**, добавив в это отношение внешний ключ **КодГруппы**.

В соответствии с **правилом 6** для реализации связи **Студент-дисциплина** создадим новое отношение **Оценки**, содержащее в качестве внешних ключей первичные ключи отношений **Студенты** и **Дисциплины**. Добавим в это отношение атрибут связи **Оценка** для регистрации результатов экзаменов, сдаваемых студентами по различным дисциплинам.

# Реляционная модель БД



## Избыточное дублирование данных и аномалии

Следует различать не избыточное и избыточное дублирование данных.

Пример неизбыточного дублирования данных:

Сотрудник	Телефон
-----------	---------

Иванов	3721
--------	------

Петров	4328
--------	------

Сидоров	4328
---------	------

Егоров	4328
--------	------

## **Пример избыточного дублирования:**

### **а) Сотрудник Телефон Номер\_комнаты**

<b>Иванов</b>	<b>3721</b>	<b>109</b>
<b>Петров</b>	<b>4328</b>	<b>111</b>
<b>Сидоров</b>	<b>4328</b>	<b>111</b>
<b>Егоров</b>	<b>4328</b>	<b>111</b>

**Здесь несколько служащих имеют один и тот же номер комнаты – отсюда и избыточность.**

**Исключение избыточности достигается переходом к двум отношениям:  
Телефон-Комнаты и Сотрудник-Комнаты:**

### **Телефон-Комнаты**

**Телефон Н\_комн**

**3721 109**

**4328 111**

### **Сотрудник-Комнаты**

**Сотрудник Н\_комн**

**Иванов 109**

**Петров 111**

**Сидоров 111**

**Егоров 111**

# Пример нежелательной избыточности

<u>Номер зачетки</u>	ФИО	Группа	Староста	Куратор
20т4	Иванов	Бин1202	Рябов	Фокин
20т5	Петров	Бин203	Сизов	Кузнецов
20т6	Сидоров	Бин1202	Рябов	Фокин

**Избыточное дублирование данных создает проблемы при обработке записей отношения, названные Э. Коддом «аномалиями».**

**Аномалиями будем называть такую ситуацию в таблицах БД, которая приведет к противоречиям в БД либо существенно усложняет обработку данных.**

**Выделяют три основных вида аномалий:**

- аномалии модификации (или редактирования);**
- аномалии удаления;**
- аномалии добавления.**

**Аномалии модификации проявляются в том, что изменение значения одного данного может повлечь за собой просмотр всей таблицы и соответствующее изменение некоторых других записей таблицы.**

**Аномалии удаления состоят в том, что при удалении какого-либо данного из таблицы может пропасть и другая информация, которая не связана напрямую с удалением данным.**

**Так, например, удаление записи о сотруднике Иванове в отношении Сотрудник-Телефон-Н\_комнаты приводит к исчезновению информации о номере телефона в 109 комнате.**

**Аномалии добавления возникают в случаях, когда информацию в таблицу нельзя поместить до тех пор, пока она не полная, либо вставка новой записи требует дополнительного просмотра таблицы.**

**Добавление записи о номере комнаты и телефоне может привести к неопределенности в случае, пока никто из сотрудников не помещен в эту комнату.**

**Средством исключения избыточности в отношениях и, как следствие, аномалий является нормализация отношений.**

## Вопросы по теме «Логическое проектирование БД»

1. Правила преобразования сущностей.
2. Правила преобразования связей. Степень связи 1:1. КП обеих сущностей обязательный.
3. Правила преобразования связей. Степень связи 1:1. КП одной сущности обязательный, другой – необязательный.
4. Правила преобразования связей. Степень связи 1:1. КП обеих сущностей необязательный.
5. Правила преобразования связей. Степень связи 1:M. КП сущности на стороне много – обязательный.
6. Правила преобразования связей. Степень связи 1:M. КП сущности на стороне много – необязательный.
7. Правила преобразования связей. Степень связи M:M.
8. Избыточное и не избыточное дублирование. Примеры.
9. Пример исключения избыточности.
10. Что такое аномалии в БД? Виды. В чем проявляются.

# Нормализация отношений

Для устранения избыточности данных и исключения аномалий применяется **нормализация отношений** – формальный метод анализа отношений на основе их первичных или потенциальных ключей и существующих функциональных зависимостей между атрибутами. При нормализации отношений они перегруппируются (производится **декомпозиция отношений**), так как при неправильной их группировке некоторые функциональные зависимости между атрибутами могут оказаться нежелательными из-за аномалий, которые они вызывают.

Нормализация предназначена для приведения структуры БД к виду, обеспечивающему **минимальную логическую избыточность**, и не имеет целью увеличение производительности обработки данных или уменьшение физического объема данных. Конечной целью нормализации является **уменьшение потенциальной противоречивости** хранимой в БД информации.

# Метод нормальных форм

Первая нормальная форма – 1НФ

Вторая нормальная форма – 2НФ

Третья нормальная форма – 3НФ

Нормальная форма Бойса-Кодда – НФБК

Четвертая нормальная форма – 4НФ

Пятая нормальная форма – 5НФ

# Функциональные зависимости

**СОТРУДНИК (ФИО, Должность, Оклад)**

**Должность зависит от ФИО**

**Оклад зависит от Должность**

**Оклад транзитивно зависит от ФИО**

# Функциональные зависимости между атрибутами

Дадим определение функциональной зависимости между атрибутами. Атрибут В функционально зависит от атрибута А, если каждому значению атрибута А соответствует в точности одно значение атрибута В. Например, в некотором отношении БД имеется функциональная зависимость атрибута ДОЛЖНОСТЬ от атрибута ФАМИЛИЯ СОТРУДНИКА.

Частичной функциональной зависимостью называется зависимость неключевого атрибута от части составного ключа. Рассмотрим пример. Пусть имеется отношение

НОМЕР ТОВАРА	НОМЕР ПОСТАВЩИКА ТОВАРА	НАИМЕНОВАНИЕ ПОСТАВЩИКА ТОВАРА	КОЛИЧЕСТВО ПОСТАВЛЯЕМОГО ТОВАРА
16	10	ООО Альянс	1000
18	20	ООО Визит	1500
16	20	ООО Визит	1520
17	30	КБ	1900

В данном отношении ключ составной: НОМЕР ТОВАРА + НОМЕР ПОСТАВЩИКА. При этом НАИМЕНОВАНИЕ ПОСТАВЩИКА ТОВАРА зависит только от части ключа – от атрибута НОМЕР ПОСТАВЩИКА ТОВАРА.

Полной функциональной зависимостью атрибута В от атрибута А называется такая зависимость, при которой атрибут В зависит от атрибута А и не зависит от любого подмножества А. Например, в вышеприведенном отношении атрибут КОЛИЧЕСТВО ПОСТАВЛЯЕМОГО ТОВАРА полно зависит от всего составного ключа Номер товара + Номер поставщика и не зависит, например, только от НОМЕРА ТОВАРА или от НОМЕРА ПОСТАВЩИКА ТОВАРА.

Атрибут С зависит от атрибута А транзитивно, если атрибут В зависит от атрибута А и атрибут С зависит от атрибута В, но обратная зависимость отсутствует. Например, в некотором отношении транзитивной зависимостью связаны атрибуты ФАМИЛИЯ ДОЛЖНОСТЬ и ОКЛАД. Т.е. атрибут ДОЛЖНОСТЬ (В) зависит от атрибута ФАМИЛИЯ (А), а атрибут ОКЛАД (С) зависит от атрибута ДОЛЖНОСТЬ (В). При этом нет обратной зависимости.

# Первая нормальная форма

## (1НФ)

Процесс нормализации отношения всегда начинается с приведения его в первую нормальную форму (1НФ) или с установки того факта, что оно уже находится в 1НФ.

**Отношение находится в 1НФ, если на пересечении любой строки и любого столбца отношения всегда находится *единственное атомарное значение*.** Другими словами, значения в домене каждого атрибута отношения не являются ни списками, ни множествами значений.

Определить понятие атомарности трудно. Значение, атомарное в одном приложении, может быть неатомарным в другом. Можно руководствоваться общим принципом, что значение не атомарно, если в приложении оно используется по частям

# Пример приведения отношения в 1НФ

Имя	Дата рождения
Анна	5 марта 1986
Александр	25 января 1987
Ольга	1 ноября 1987
Федор	14 сентября 1986

Имя	День рождения	Месяц рождения	Год рождения
Анна	5	Март	1986
Александр	25	Январь	1987
Ольга	1	Ноябрь	1987
Федор	14	Сентябрь	1986

# Пример приведения отношения в 1НФ

Имя	Пол
{Александр, Федор}	Мужской
Ольга	Женский

Имя	Пол
Александр	Мужской
Федор	Мужской
Ольга	Женский

## Процесс нормализации

Различают первую нормальную форму (1НФ), вторую (2НФ), третью (3НФ), нормальную форму Бойеса-Кодда (НФБК), четвертую (4НФ) и пятую (5НФ). В большинстве случаев при проектировании БД достаточно первых трех нормальных форм.

*Отношение находится в 1НФ, если на пересечении любой строки и любого столбца отношения всегда находится единственное атомарное значение.*

Приведем пример нарушения 1НФ.

Отношение КЛУБ

ПОЧТА

ПА-  
РОЛЬ

ИМЯ

ТЕЛЕ-  
ФОН

[dsa@mail.ru](mailto:dsa@mail.ru)

Gmal

Иванов  
А.

234-77-45

[er@yandex.ru](mailto:er@yandex.ru)

YUrtt

Сазонов  
П.Р.

345-34-77  
123-89-22

[nn@mail.ru](mailto:nn@mail.ru)

Treo1

Петров  
А.

222-12-44

.

.

**У пользователя Сазонова имеется два телефона, что приводит к аномалиям разного рода. Поэтому желательно отношение КЛУБ разбить на два:**

<b>ПОЧТА</b>	<b>ПАРОЛЬ</b>	<b>ИМЯ</b>
<b>dsa@mail.ru</b>	<b>Gmal</b>	<b>Иванов А.</b>
<b><u>er@yandex.ru</u></b>	<b>YUrtt</b>	<b>Сазонов П.Р.</b>
<b><u>nn@mail.ru</u></b>	<b>Treo1</b>	<b>Петров А.</b>

.  
.

<b>ПОЧТА_ИМЯ</b>	<b>ТЕЛЕФОН</b>
<b>dsa@mail.ru</b>	<b>234-77-45</b>
<b><u>er@yandex.ru</u></b>	<b>345-34-77</b>
<b><u>er@yandex.ru</u></b>	<b>123-89-22</b>
<b><u>nn@mail.ru</u></b>	<b>222-12-44</b>

.  
.