**Лекция 3.** Проектирование реляционных БД с использованием нормализации

**Цель лекции:** Ознакомиться с достоинствами и недостатками реляционного подхода к построению инфологической модели. Понять, как влияет *нормализация* отношений на работу *базы данных*. Понять различия между простым и *составным ключом*.

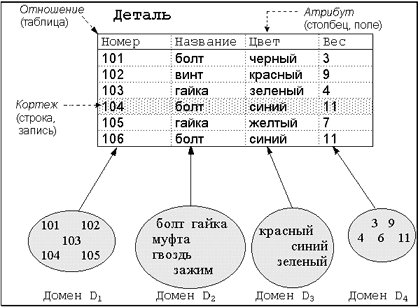
**Реляционная модель данных**

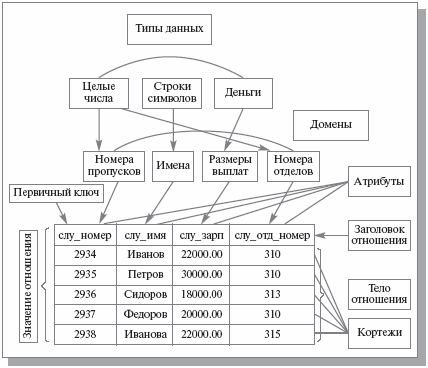
Реляционная модель есть *представление* *БД* в виде совокупности упорядоченных нормализованных отношений.

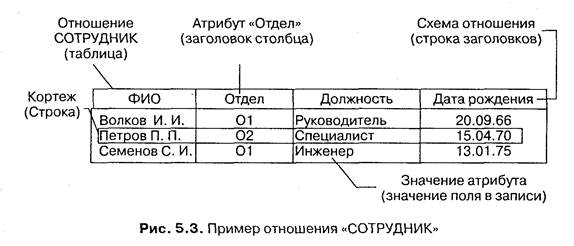
КОРТЕЖ – это элемент отношения. Попросту говоря, кортеж - это набор именованных значений заданного типа.

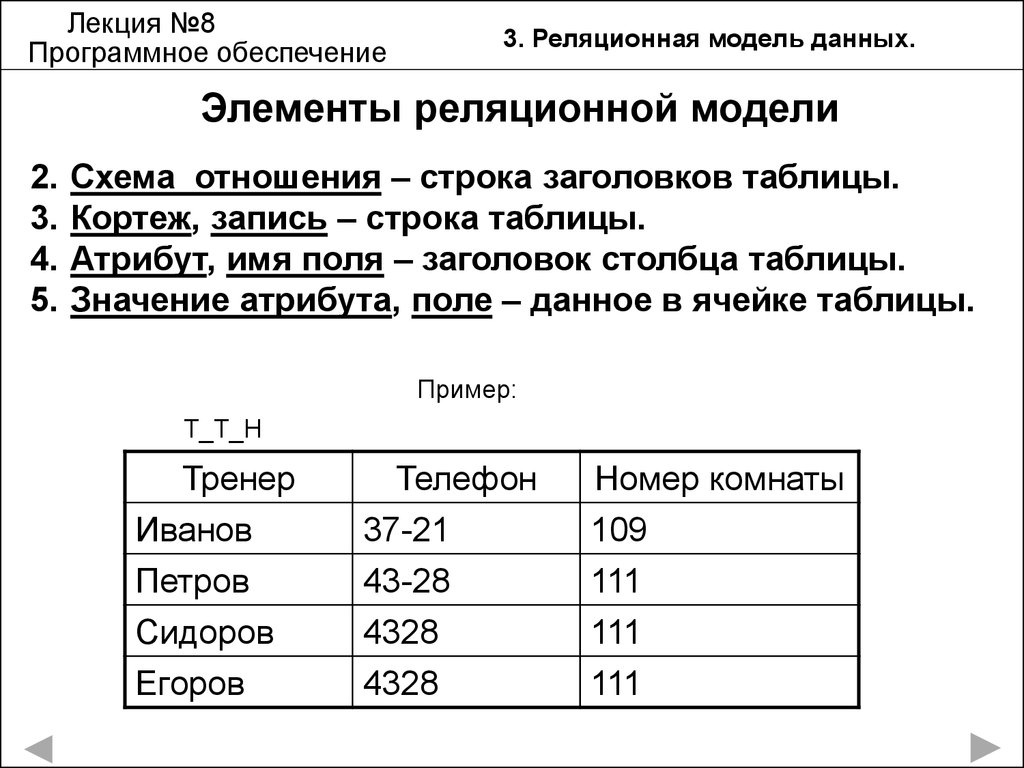
Домен - это "вид" данных, которые может содержать данный атрибут. Более четкое определение -  **это набор всех допустимых значений, которые может содержать данный атрибут.**

Примеры:









Для реляционных отношений характерны следующие особенности:

1. Любой *тип записи* содержит только простые (по структуре) элементы данных.
2. Порядок кортежей в таблице несуществен.
3. Упорядочение значащих атрибутов в кортеже должно соответствовать упорядочению атрибутов в реляционном отношении.
4. Любое отношение должно содержать один атрибут или более, которые вместе составляют уникальный первичный ключ.
5. Если между двумя реляционными отношениями существует зависимость, то одно отношение является исходным, второе - подчиненным.
6. Чтобы между двумя реляционными отношениями существовала зависимость, атрибут, служащие первичным ключом в исходном отношении, должны также присутствовать в подчиненном отношении.

**Пример 5.1.** Представим *БД* "Учебный процесс" в виде реляционной модели ([таблица 5.1](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#table.5.1)).

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 5.1. | | | |
| **а) Отношение ГРУППА** | | | |
| **Индекс ИГ** | **Название группы НГ** | **Количество ответов КОЛ** | **Проходной балл ПБАЛЛ** |
| 1 | А1 | 16 | 4,3 |
| 2 | А2 | 28 | 4,0 |
| 3 | А3 | 18 | 4,3 |
| **б) Отношение СТУДЕНТ** | | | |
| **Номер зачетной книжки НЗ** | **ИГ** | **Фамилия И.О. СФИО** | **Год рождения ГР** |

**Понятие реляционный** (англ. *relation* - *отношение*) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и *реляционного исчисления* для обработки данных.

**Реляционная модель ориентирована на организацию данных в виде двумерных таблиц.**

**Каждая *реляционная таблица* представляет собой*****двумерный массив* и обладает следующими свойствами:**

* каждый элемент таблицы - один элемент данных;
* все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
* каждый столбец имеет уникальное имя;
* одинаковые строки в таблице отсутствуют;
* порядок следования строк и столбцов может быть произвольным.

**Пример 5.2.** Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе ([таблица 5.2](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#table.5.2)).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица 5.2. Пример реляционной таблицы | | | | | |
| **№ личного дела** | **Фамилия** | **Имя** | **Отчество** | **Дата рождения** | **Группа** |
| 16493 | Сергеев | Петр | Михайлович | 01.01.76 | ИСТ 11 |
| 16593 | Петрова | Анна | Владимировна | 15.03.75 | СК 12 |
| 16693 | Анохин | Андрей | Борисович | 14.04.76 | ИСТ 11 |

**Понятие информационного объекта**

**Информационный объект** - это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных *реквизитов* (информационных элементов). Такими сущностями для информационных объектов могут служить: цех, склад, материал, вуз, студент, сдача экзаменов и т.д.

Информационный *объект* определенного реквизитного состава и структуры образует *класс* (тип), которому присваивается уникальное имя (символьное обозначение), например Студент, Сессия, Стипендия.

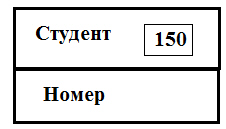
Информационный *объект* имеет множество реализации - экземпляров, каждый из которых представлен совокупностью конкретных значений реквизитов и идентифицируется значением ключа (простого - один *реквизит* или составного - несколько реквизитов). Остальные реквизиты информационного объекта являются описательными. При этом одни и те же реквизиты в одних информационных объектах могут быть ключевыми, а в других- описательными. Информационный *объект* может иметь несколько ключей.

**Пример 5.3** В [таблице 5.2](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#table.5.2) представлен пример структуры и экземпляров информационного объекта Студент.

В информационном объекте Студент ключом является *реквизит* **Номер** (№ личного дела), к описательным реквизитам относятся: *Фамилия*(Фамилия студента), *Имя* (Имя студента), *Отчество* (Отчество студента), *Дата* (Дата рождения), *Группа* (№ группы). Если отсутствует *реквизит* **Номер**, то для однозначного определения характеристик конкретного студента необходимо использование *составного ключа* из трех реквизитов: *Фамилия + Имя + Отчество*.

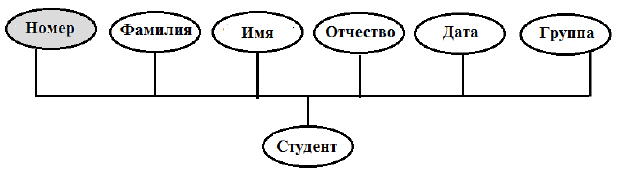
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Таблица 5.2. Пример структуры и экземпляров информационного объекта | | | | | | |
| **Структура** | **Номер** | **Фамилия** | **Имя** | **Отчество** | **Дата** | **Группа** |
| **Экземпляры инф.объекта Студент** | 16493 | Сергеев | Петр | Михайлович | 01.01.96 | ИСТ 11 |
| 16593 | Петрова | Анна | Викторович | 15.03.95 | СК 12 |
| 16693 | Анохин | Роман | Борисович | 14.04.96 | ИСТ 11 |

**Пример 5.4** На [рис. 5.1](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#image.5.1) изображен пример компактного представления информационного объекта Студент с обозначением имени объекта, ключа и указанием максимально возможного числа *экземпляров записи*.



**Рис. 5.1.**Пример компактного представления информационного объекта

**Пример 5.5** Пример представления информационного объекта Студент в виде графа на [рис. 5.2](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#image.5.2).



**Рис. 5.2.**Пример представления информационного объекта в виде графа

**Нормализация отношений**

**Понятие нормализации отношений**

Одни и те же данные могут группироваться в таблицы (отношения) различными способами, т.е. возможна организация различных наборов отношений взаимосвязанных информационных объектов. Группировка атрибутов в отношениях должна быть рациональной, т.е. С ЦЕЛЬЮ минимизации *дублирования данных* и упрощения процедуры их обработки и обновления.

Определенный набор отношений обладает лучшими свойствами при включении, модификации, удалении данных, чем все остальные возможные наборы отношений, если он отвечает требованиям нормализации отношений.

**Нормализация отношений** - ограничения на формирование отношений (таблиц), которое позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение (ввод, корректировку) базы данных.

Е. Коддом выделены три **нормальные формы отношений** и предложен механизм, позволяющий любое отношение преобразовать к третьей (самой совершенной) *нормальной форме*.

**Первая нормальная форма**

Отношение называется нормализованным или приведенным к **первой нормальной форме**, если все его атрибуты простые (данные неделимы).

Преобразование отношения к первой *нормальной форме* может привести к увеличению количества реквизитов (полей) отношения и изменению ключа.

Например, отношение Студент = (*Номер*, Фамилия, Имя, Отчество, Дата, Группа) находится в первой *нормальной форме*.

**Вторая нормальная форма**

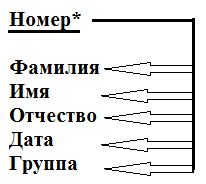
Чтобы рассмотреть вопрос приведения отношений ко *второй нормальной форме*, необходимо дать пояснения к таким понятиям, как *функциональная зависимость* и полная *функциональная зависимость*.

Описательные реквизиты информационного объекта логически связаны с общим для них ключом, эта связь носит характер **функциональной зависимости** реквизитов.

**Функциональная зависимость** реквизитов - зависимость, при которой в экземпляре информационного объекта определенному значению ключевого реквизита соответствует только одно значение описательного реквизита.

Такое определение *функциональной зависимости* позволяет при анализе всех взаимосвязей реквизитов предметной области выделить самостоятельные информационные объекты.

**Пример 5.6** Пример графического изображения функциональных зависимостей реквизитов Студент показан на [рис. 5.3](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#image.5.3), на котором ключевой *реквизит* указан \*.



**Рис. 5.3.**Графическое изображение функциональной зависимости реквизитов

В случае *составного ключа* вводится понятие **функционально полной** зависимости.

**Функционально полная зависимость** неключевых атрибутов заключается в том, что каждый неключевой атрибут функционально зависит от ключа, но не находится в *функциональной зависимости* ни от какой части *составного ключа*.

Таблица находится во **второй нормальной форме**, если она удовлетворяет требованиям первой *нормальной формы* и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом, то есть любое не *ключевое поле* однозначно идентифицируется полным набором *ключевых полей*.

Итак, таблица, находящаяся во *второй нормальной форме*, должна удовлетворять следующим правилам:

* таблица должна содержать данные об одном типе объектов;
* каждая таблица должна содержать одно поле или несколько полей, образующих уникальный идентификатор (или первичный ключ) для каждой строки;
* все поля, не имеющие ключа, должны определяться полным уникальным идентификатором данной таблицы.

**Пример 5.7**

|  |  |
| --- | --- |
| **Отношение Студент = (Номер, Фамилия, Имя, Отчество, Дата, Группа)** | находится в первой и во *второй нормальной форме* одновременно, так как описательные реквизиты однозначно определены и функционально зависят от ключа **Номер**. |

|  |  |
| --- | --- |
| **Отношение Успеваемость =**  **(Номер, Фамилия, Имя, Отчество, Дисциплина, оценка)** | находится в первой *нормальной форме* и имеет составной ключ **Номер + Дисциплина**. Это отношение не находится во *второй нормальной форме*, так как атрибуты Фамилия, Имя, Отчество не находятся в полной *функциональной зависимости* с составным *ключом отношения*. |

**Третья нормальная форма**

Понятие *третьей нормальной формы* основывается на понятии **нетранзитивной** зависимости.

**Транзитивная зависимость** наблюдается в том случае, если один из двух описательных реквизитов зависит от ключа, а другой описательный *реквизит* зависит от первого описательного реквизита.

Таблица находится **в третьей нормальной форме**, если она удовлетворяет определению второй *нормальной формы* и ни одно из ее неключевых полей функционально не зависит от любого другого неключевого поля. Можно сказать, что таблица находится в *третьей нормальной форме*, если она находится во *второй нормальной форме* и каждое неключевое поле нетранзитивно зависит от первичного ключа.

Требование *третьей нормальной формы* сводится к тому, чтобы все не ключевые поля зависели только от первичного ключа и не зависели друг от друга. Другими словами, нужно иметь возможность изменять значение любого не ключевого поля, не изменяя значения любого другого поля базы данных. Это требование исключает любое поле, значения в котором получаются как результат вычислений, использующих значения других полей.

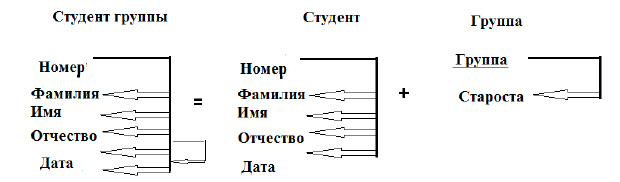
**Пример 5.8** Если в состав описательных реквизитов информационного объекта **Студент** включить *фамилию старосты группы* (**Староста**), которая определяется только номером группы, то одна и та же фамилия старосты будет многократно повторяться в разных экземплярах данного информационного объекта.

В этом случае наблюдаются *затруднения в корректировке фамилии* старосты в случае назначения нового старосты, а также *неоправданный расход памяти* для хранения дублированной информации.

Для устранения транзитивной зависимости описательных реквизитов необходимо провести "**расщепление**" исходного информационного объекта. В результате расщепления часть реквизитов удаляется из исходного информационного объекта и включается в состав других (возможно, вновь созданных) информационных объектов.

**Пример 5.9**  "Расщепление" информационного объекта, содержащего *транзитивную* зависимость описательных реквизитов, показано на [рис. 5.4](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#image.5.4). Как видно из [рис. 5.4](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=1#image.5.4), исходный информационный объект Студент группы представляется в виде совокупности правильно структурированных информационных объектов (**Студент и Группа**), реквизитный состав которых тождественен исходному объекту.

Отношение Студент = (**Номер**, Фамилия, Имя, Отчество, Дата, Группа) находится одновременно в первой, второй и *третьей нормальной форме*.



**Рис. 5.4.**Расщепление" информационного объекта, содержащего транзитивную зависимость описательных реквизитов

**Типы связей. Свойства отношений**

**Свойства отношений:**

**1. Отсутствие кортежей-дубликатов**. Из этого свойства вытекает наличие у каждого кортежа первичного ключа. *Для каждого отношения, по крайней мере, полный набор его атрибутов является первичным ключом*. Однако, при определении первичного ключа должно соблюдаться требование "минимальности", т.е. в него не должны входить те атрибуты, которые можно отбросить без ущерба для основного свойства первичного ключа - однозначно определять *кортеж*.

**2. Отсутствие упорядоченности кортежей.**

**3. Отсутствие упорядоченности атрибутов**. Для ссылки на *значение* атрибута всегда используется имя атрибута.

**4. Атомарность значений атрибутов**, т.е. среди значений домена не могут содержаться *множества* *значений (отношения*).

Свойства реляционных отношений:

* **В отношении нет одинаковых кортежей**
* **Кортежи не упорядочены (сверху вниз)**
* **Атрибуты не упорядочены (слева направо)**
* **Все значения атрибутов атомарны**

Реляционные *базы данных* состоят из нескольких таблиц, *cвязь* между которыми устанавливается с помощью совпадающих полей. Каждая *запись* в таблицах идентифицирует один *объект*. О*тношение* между объектами определяет *отношение* между таблицами.

Существует 4 типа отношений:

* **Отношение "один-к-одному"** (1:1) означает, что каждая запись в одной таблице соответствует только одной записи в другой таблице.
* **Отношение "один-ко-многим"** (1 :М) означает, что каждой записи в одной таблице соответствует одна или несколько записей в другой таблице.
* **Отношение "многие-к-одному"** (М:1) аналогично рассмотренному ранее типу "один-ко-многим". Тип отношения между объектами зависит от вашей точки зрения.
* **Отношение "многие-ко-многим"** (М:М). возникает между двумя таблицами в тех случаях, когда каждой запись в одной таблице соответствует 0, 1, 2 и более записей в другой таблице и наоборот.

В большинстве случаев любые две таблицы связаны отношением **"один-ко-многим"**. Это означает, что любая *запись* в первой таблице может быть связана с несколькими записями во второй, однако любая *запись* второй таблицы связана только с одной записью в первой.

**Пример 5.10** Дана совокупность информационных объектов, отражающих учебный процесс в вузе:

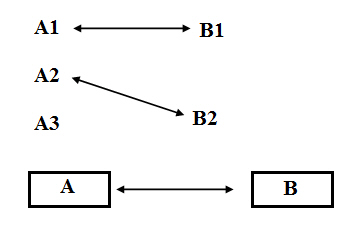
СТУДЕНТ (*Номер*, Фамилия, Имя, Отчество, Пол, Дата рождения, *Группа*)

*СЕССИЯ* (*Номер*, Оценка1, Оценка2, ОценкаЗ, Оценка4, Результат)

СТИПЕНДИЯ (*Результат*, *Процент*)

ПРЕПОДАВАТЕЛЬ (*Код преподавателя*. Фамилия, Имя, Отчество)

**Связь один к одному** (1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот. [Рисунок 5.5](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=2#image.5.5) иллюстрирует указанный тип отношения.

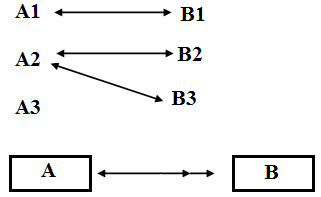


**Рис. 5.5.**Графическое изображение реального отношения 1:1

**Пример 5.11** Примером связи 1:1 может служить *связь* между информационными объектами СТУДЕНТ и СЕССИЯ:

СТУДЕНТ <-> СЕССИЯ Каждый студент имеет определенный набор экзаменационных оценок в сессию.

При связи **один ко многим** (1:М) одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А. Графически данное соответствие имеет вид, представленный на [рис. 5.6](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=2#image.5.6).



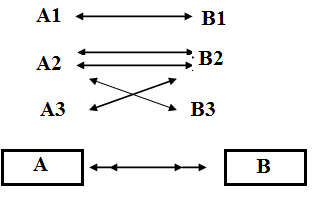
**Рис. 5.6.**Графическое изображение реального отношения 1:М

**Пример 5.12** Примером связи 1 :М служит *связь* между информационными объектами СТИПЕНДИЯ и СЕССИЯ:

СТИПЕНДИЯ <-" СЕССИЯ

Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

*Связь* **многие ко многим** (М:М) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В и наоборот. На [рис. 5.7](https://www.intuit.ru/studies/courses/3439/681/lecture/14024?page=2#image.5.7) графически представлено указанное соответствие.



**Рис. 5.7.**Графическое изображение реального отношения М:М

**Пример 5.13** Примером данного отношения служит *связь* между информационными объектами СТУДЕНТ и ПРЕПОДАВАТЕЛЬ:

СТУДЕНТ "-" ПРЕПОДАВАТЕЛЬ

Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

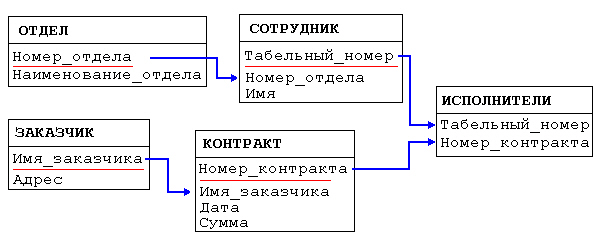
**Простые и составные ключи**

*Первичный ключ* может состоять из единственного поля таблицы, значения которого уникальны для каждой записи. Так, например, на предприятии не может быть двух работников с одинаковыми табельными номерами, поэтому в таблице, содержащей записи о работниках, табельный номер может быть первичным ключом. Такой *первичный ключ* называют простым ключом.

Если *таблица* не имеет единственного уникального поля, *первичный ключ* может быть составлен из нескольких полей, совокупность значений которых гарантирует уникальность. Так, имя, фамилия, отчество, номер паспорта, серия паспорта не могут быть первичными ключами по отдельности, так как могут оказаться одинаковыми у двух и более людей. Но не бывает двух личных документов одного типа с одинаковыми серией и номером. Поэтому в таблице, содержащей записи о людях, первичным ключом может быть набор полей, состоящий из типа личного документа, его серии и номера. Такой *первичный ключ* называют *составным ключом*

Все остальные *ключи отношения* называются *возможными ключами*.

В отличие от иерархической и *сетевой моделей данных* в реляционной отсутствует понятие группового отношения. Для отражения ассоциаций между кортежами разных отношений используется дублирование их ключей. Рассмотренный иерархический и сетевой пример *базы данных*, содержащей сведения о подразделениях предприятия и работающих в них сотрудниках, применительно к реляционной модели будет иметь вид:



База данных о подразделениях и сотрудниках предприятия

Например, *связь* между отношениями ОТДЕЛ и СОТРУДНИК создается путем копирования первичного ключа "Номер\_отдела" из первого отношения во второе. Таким образом:

* для того, чтобы получить список работников данного подразделения, необходимо из таблицы ОТДЕЛ установить значение атрибута "Номер\_отдела", соответствующее данному "Наименованию\_отдела" выбрать из таблицы СОТРУДНИК все записи, значение атрибута "Номер\_отдела" которых равно полученному на предыдущем шаге.
* для того, чтобы узнать в каком отделе работает сотрудник, нужно выполнить обратную операцию:
  + определяем "Номер\_отдела" из таблицы СОТРУДНИК
  + по полученному значению находим запись в таблице ОТДЕЛ.

Атрибуты, представляющие собой копии ключей других отношений, называются внешними ключами.

Иногда возникает потребность разбить одну таблицу на более мелкие - проблема может заключаться в том, что некоторые сведения из нее используются не слишком часто, или в том, что какие-то данные не предназначаются для всеобщего доступа. Например, часть информации о факультетах нужна только для рекламных целей и используется очень редко. С другой стороны, сведения о заработной плате должны быть доступны только определенным сотрудникам. В любом из этих случаев можно создать отдельную таблицу и связать ее с исходной таблицей отношением типа "*один-к-одному*". Это означает, что любая *запись* в первой таблице связана только с одной записью во второй.

Если же между таблицами необходимо организовать *связь* "*многие-ко-многим*", то придется создать дополнительную таблицу пересечения, с помощью которой одна *связь* будет сведена к двум связям типа "*один-ко-многим*".

**Вопросы для самопроверки**

1. Назовите характерные особенности реляционных отношений.
2. На что ориентирована реляционная модель?
3. Какими свойствами обладает реляционная таблица?
4. Понятие информационного объекта.
5. Понятие нормализации отношений.
6. *Первая**нормальная форма*.
7. *Вторая нормальная форма*.
8. *Третья нормальная форма*.
9. Понятие простого ключа.
10. Понятие *составного ключа*.
11. Какие *типы связей* Вы знаете, охарактеризуйте их.