

2.2. Структуры

Общеприняты следующие понятия, определяющие элементарные единицы данных и интерпретаций (не путать с атомарной единицей информации).

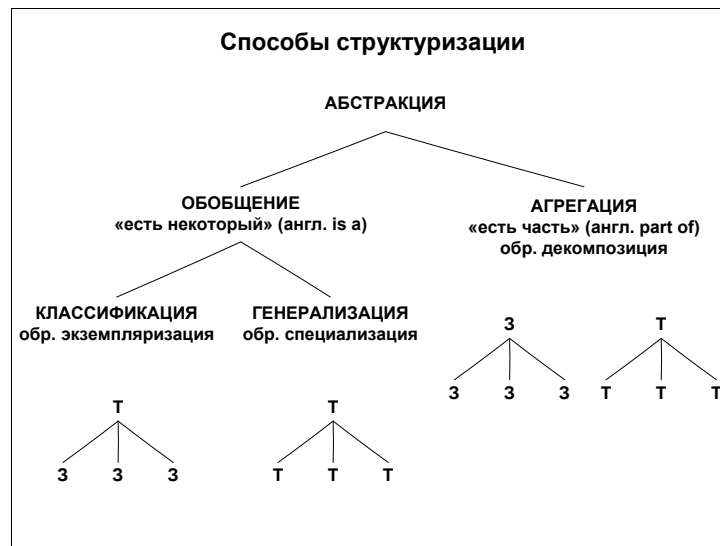
В общем случае знак – это материальный предмет (явление, событие), выступающий в качестве представителя некоторого другого предмета, свойства или отношения и используемый для приобретения, хранения, переработки и передачи сообщений (информации, знаний). В контексте технологии БД можно дать следующие определения.

Определение 2.2.1. Знак – это данное (строка символов, число, дата или их агрегат), определяющее конкретный объект, связь объектов или значение характеристики объекта.

Определение 2.2.2. Тип – это именованный класс подобных знаков. Имя типа определяет интерпретацию принадлежащих типу знаков.

Структурный компонент схемы БД представляет собой определения типов (типов объектов ПрО, типов их связей и их характеристик). Типы в схеме не изолированы друг от друга, а образуют взаимосвязанную систему типов. Задачей структуризации ПрО в конкретной модели данных и является построение такой системы типов с использованием понятий, предлагаемых правилами структуризации данных этой модели.

В этом параграфе мы познакомимся с основными мыслительными процессами, сопутствующими структуризации, основными формами данных, использующимися в различных моделях данных, а также не забудем обсудить, как осуществляется интерпретация данных в этих формах.



Одним из основных способов структуризации является абстракция. Общеизвестны следующие варианты содержания понятия абстракции.

Определение 2.2.3. Абстракция (от латинского abstractio – отвлечение):

1) Метод научного исследования, основанный на том, что при изучении некоторого явления, процесса не учитываются его несущественные стороны и признаки; это позволяет упрощать картину изучаемого явления и рассматривать его как бы в «чистом виде».

2) Продукт познания (понятие, описание, закон, модель, идеальный объект и т. п.), рассмотренный в сопоставлении с конкретной эмпирической действительностью, которая не фиксируется в этом продукте во всем богатстве своих свойств и связей.

3) Познавательная деятельность – процесс абстрагирования, – направленная на получение абстракции.

Абстракция, как предмет, представляет собой совокупность деталей конкретного явления или группы явлений, которая может быть соответствующим образом рассмотрена и поименована, как самостоятельное целое. Абстракция, как деятельность, предполагает, что несущественные частные детали должны быть опущены, а внимание должно быть сосредоточено на важных общих особенностях.

В моделировании данных абстракция используется для образования категорий данных. Кроме того, абстракция позволяет на основе одних категорий построить другие, более общие.

В технологии БД, как и вообще, в информатике, выделяют два вида абстракции: обобщение и агрегацию.

Определение 2.2.4. Обобщение – это абстракция, которая позволяет соотнести множество знаков или типов с одним общим типом в соответствии с отношением «есть некоторый» (англ. is a). В зависимости от вида объектов обобщения рассматривают два частных случая обобщения – классификацию и генерализацию. **Классификация** – это обобщение знаков до типа (отношение «экземпляр-класс»), а **генерализация** – это обобщение типов до типа (отношение «подкласс-суперкласс»).

Обобщенный тип обладает всеми признаками, общими для базовых знаков или типов. Иначе говоря, все признаки обобщенного типа могут быть унаследованы базовыми типами.

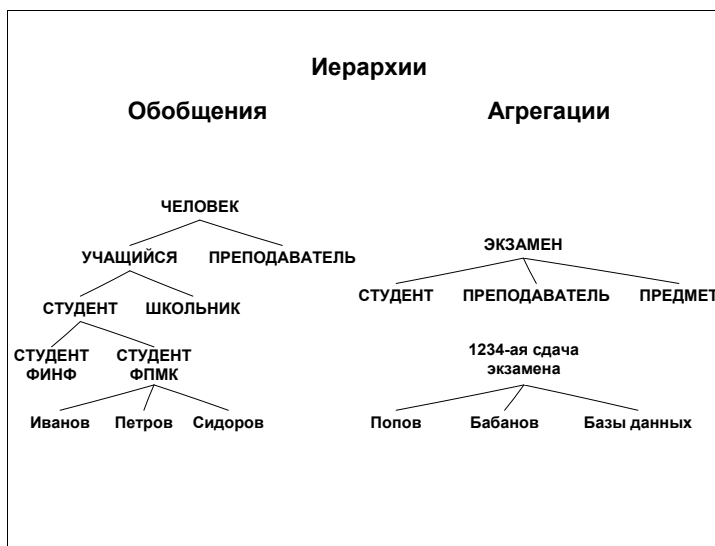
Определение 2.2.5. Агрегация – это абстракция, посредством которой объект конструируется из других, базовых объектов в соответствии с отношением «есть часть» (англ. part of). Агрегация возможна как на уровне знаков (сложный знак собирается из более простых знаков), так и на уровне типов (сложный тип собирается из более простых

типов). Более того, агрегация на уровне типов предполагает множество агрегаций на уровне знаков.

Приведенные способы мышления используются при синтезе – переходе от простого к сложному, от частного к общему. Поскольку задачей структуризации является построение графа типов, можно идти к нему и методами анализа, т.е. используя обратные процессы. Операциями, обратными классификации, генерализации и агрегации, являются соответственно операции экземпляризации, специализации и декомпозиции.

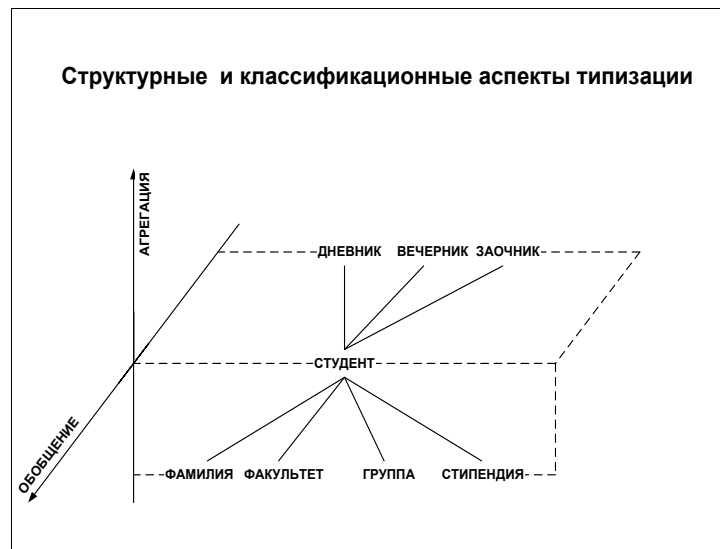
На слайде представлено не просто дерево, а иерархия обобщения способов структуризации. Все они являются разновидностями абстракции, поэтому это понятие образует «корень» иерархии. Двумя специализациями абстракции являются понятия «обобщение» и «агрегация». И, наконец, классификация и генерализация – частные случаи обобщения.

Маленькие двухуровневые деревья, сопоставленные каждому «листу» иерархии обобщения способов структуризации, демонстрируют, для каких элементов (знаков – «З» или типов – «Т») могут применяться соответствующие операции, и какой элемент образуется в результате.



Последовательно применяя обобщения и агрегации, можно строить соответствующие многоуровневые иерархии. В левой части слайда приведена иерархия обобщения. На ее нижнем уровне представлены знаки, обозначающие конкретные предметы ПрО, на втором – обобщающие их типы, на третьем – типы, обобщающие эти типы и т.д. Для упрощения картины дерево приведено не полностью, некоторые ветви не раскрыты. На первой фазе обобщения мы используем классификацию, на всех последующих – генерализацию.

В правой части рисунка приведены две двухуровневые иерархии агрегации: нижняя касается знаков, верхняя – соответствующих им типов. Первая показывает, что знак, представляющий конкретный (в данном случае 1234-й) факт сдачи экзамена рассматривается как агрегат, составленный из знака «*Попов*», указывающего на студента, знака «*Бабанов*», соответствующего преподавателю-экзаменатору, и знака «*Базы данных*», определяющего сдаваемую дисциплину. Очевидно, что общую структуру связей типа *ЭКЗАМЕН* задает аналогичная агрегация на уровне типов, приведенная выше.



Агрегация и обобщение применяются взаимодополняющим образом и выражают структурные и классификационные аспекты типизации. Структуру типа можно представить как агрегат базовых типов, а сам агрегат, как и базовые типы, могут служить объектом обобщения. Классификация типов может быть выражена иерархией обобщения, а структура – иерархией агрегации. Эти две иерархии правомерно рассматривать как независимые (ортогональные).

Рисунок, приведенный на слайде, демонстрирует эти процессы в трехмерном пространстве: снизу вверх мы агрегируем, приближаясь с заднего плана, – обобщаем, направление слева направо используется для записи имен типов.

На рисунке приведена агрегация типов *ФАМИЛИЯ*, *ФАКУЛЬТЕТ*, *ГРУППА*, *СТИПЕНДИЯ* в тип *СТУДЕНТ*. Так как тип *СТУДЕНТ* в свою очередь есть обобщение типов *ДНЕВНИК*, *ВЕЧЕРНИК*, *ЗАОЧНИК*, последние наследуют признаки *ФАМИЛИЯ*, *ФАКУЛЬТЕТ*, *ГРУППА* и *СТИПЕНДИЯ*. Таким образом, если некоторый тип есть и обобщение, и агрегат одновременно, то его подтипы наследуют его признаки. Агрегация позволяет определить структуру объектов, а обобщение – механизм наследования признаков.

Абстракции используются для придания множествам объектов некоторой семантики.

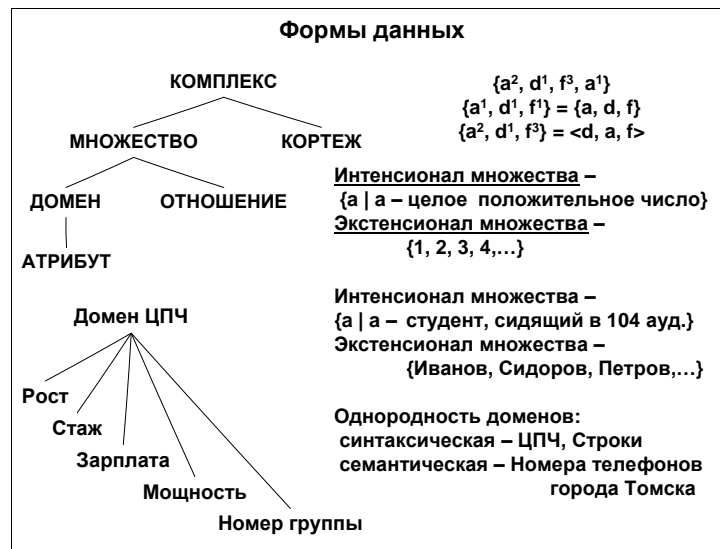
В отличие от естественного языка, в котором, как уже отмечалось, интерпретации каждый раз указываются непосредственно с данными, в технологии БД интерпретации однократно определяются в схеме и сопоставляются с множественными данными. Естественно, в схеме воплощен некий обобщенный взгляд на ПрО, часто не учитывающий несущественные детали конкретных объектов.

На первом этапе проектирования мы определяем понятия, релевантные задачам ПрО, и фиксируем их семантику, подбирая им подходящие имена (например, *СТУДЕНТ*). Далее для каждого понятия определяется необходимый для задач ПрО набор признаков, единообразно описывающий объекты данного понятия. Каждому признаку также приписывается некоторое имя, отражающее его смысл (*ФАМИЛИЯ*, *ФАКУЛЬТЕТ*, *ГРУППА*, *СТИПЕНДИЯ*).

В дальнейшем, если в ПрО обнаруживается объект, подпадающий под некоторое понятие (удовлетворяющий условию его содержания), мы строим его абстрактное представление, определяя значения фиксированных для понятия признаков. Агрегат этих значений (являющийся, кстати, знаком) обозначает в нашей системе объект ПрО, а его интерпретация задается именем соответствующего понятия или типа. Интерпретация его значений признаков осуществляется с помощью их имен. Таким образом, интерпретации,

заданные на уровне типов, позволяют приписывать семантику данным или знакам. И это позволяет осуществлять механизм абстракции.

Далее речь пойдет о традиционных формах, используемых в различных моделях для хранения и представления данных. Заявленный ранее обобщенный уровень рассмотрения, естественно, заставляет нас обратиться к математическим формам, как раз и обеспечивающим требуемый взгляд на данные. Хотя будут представлены и специфические для моделирования данных понятия.



Понятие множества, или совокупности, принадлежит к числу простейших математических понятий; оно не определяется, но может быть пояснено при помощи примеров. Так, можно говорить о множестве всех книг, составляющих данную библиотеку, множестве всех точек данной линии, множестве всех решений данного уравнения. Книги данной библиотеки, точки данной линии, решения данного уравнения являются элементами соответствующего множества.

Чтобы определить множество, достаточно указать характеристическое свойство элементов, т. е. такое свойство, которым обладают все элементы этого множества и только они. Может случиться, что данным свойством не обладает вообще ни один предмет; тогда говорят, что это свойство определяет пустое множество. То, что данный предмет x есть элемент множества M , записывают так: $x \in M$ (читают: x принадлежит множеству M). Множество может быть задано указанием общего свойства его элементов или прямым перечислением элементов.

Исходя из всего выше изложенного, можно дать следующее неформальное определение.

Определение 2.2.6. Множество – это собрание правильно идентифицированных объектов, удовлетворяющих условию принадлежности. Данное определение отражает специфику моделирования данных, в частности, в нем фигурируют «правильно идентифицированные объекты». Действительно, в других областях знаний и в человеческом воображении можно оперировать множествами произвольных объектов. Мы же говорим о формах, в которых представляются данные в знаковых системах, а значит элементы наших множеств (знаки) должны как минимум удовлетворять синтаксическим правилам задания знаков, а также, как правило, быть семантически значимыми. Это как раз и предполагает «правильная идентификация» объектов ПрО.

В классической теории множеств эта форма характеризуется следующими свойствами:

- множества необязательно должны иметь фиксированное количество элементов, они могут быть и бесконечными;
- на элементах множества отсутствует какой-либо порядок;
- элементы-дубликаты во множестве не имеют смысла.

Для правильного представления о формах данных уместен взгляд на них с точки зрения так называемой операционной семантики. В соответствии с этим подходом форма данных определяется своим интерфейсом – набором сообщений, в ответ на которые она может производить какие-то манипуляции со своими данными. Так вот, множество – это такая форма данных, которая может реагировать на следующие сообщения:

- *добавить*: «элемент» (независимо от того, принадлежал ли указанный «элемент» множеству до получения этого сообщения, после этого он считается принадлежащим множеству);
- *удалить*: «элемент» (независимо от того, принадлежал ли указанный «элемент» множеству до получения этого сообщения, после этого он считается не принадлежащим множеству);
- *есть?*: «элемент» (ответ «да», если указанный «элемент» принадлежит множеству, и «нет» – в противном случае).

Поскольку одной такой формы данных как множество недостаточно для моделирования данных, Чайлдсом (Childs D.L.) была предложена теория расширенных множеств (англ. Extended Set Theory), формально учитывающая порядок и ориентированная на проблемы моделирования данных. Основным формальным объектом, рассматриваемым в теории расширенных множеств, является комплекс.

Определение 2.2.7. Комплекс – это собрание правильно идентифицированных объектов, удовлетворяющих условию i -принадлежности. Отношение i -принадлежности между элементом x и комплексом Y (записывается $x \in iY$) наблюдается тогда и только тогда, когда x находится в i -ой позиции комплекса Y .

Таким образом, в отличие от множества элементы комплекса распределены по позициям, определяемым целыми положительными числами. Причем в каждой позиции может быть любое (в т.ч. и бесконечное) число различных элементов. В пределах одной позиции порядок элементов не определен. В разных позициях могут быть совпадающие элементы. Позиция элемента в комплексе записывается с помощью верхнего индекса.

Олицетворением комплекса в нашей жизни является блок почтовых ящиков в жилом подъезде. Каждый из них имеет номер и является не чем иным, как позицией комплекса, куда почтальон помещает периодику и корреспонденцию, причем их дубликаты, как правило, не уместны в одном ящике. Чего не скажешь о разных ящиках.

Множества и n -местные кортежи есть специальные случаи комплексов.

Определение 2.2.8. Множество – это комплекс, все элементы которого находятся в одной, предположим первой позиции. n -местный кортеж или просто **кортеж** – это комплекс, который имеет по одному элементу в каждой позиции от 1 до n .

Уместно отметить, что на слайде в левом верхнем углу приведена иерархия обобщения рассматриваемых форм данных, которая отражает отношения специализации между ними. Вверху справа приведены примеры произвольного комплекса, комплекса-множества и комплекса-кортежа. Для двух последних показаны также традиционные нотации.

Сам комплекс, в отличие от его специализаций, не является структурным понятием ни одной из моделей данных. Однако это понятие способствует правильному восприятию других, встречающихся на практике форм данных.

Определение 2.2.9. Множества, как и некоторые другие формы данных, характеризуются двумя важными свойствами. Первое из них, дефиниционное по своей природе, называется интенционалом множества. **Интенционал множества** определяет свойства, общие для всех элементов всех реализаций множества (в математике его называют условием принадлежности). Второе свойство является репрезентативным и носит название **экстенционала множества**. Он определяет актуальную реализацию множества путем явного указания его элементов. Обратите внимание, в условии интенционала фигурируют сами объекты ПрО, в экстенционале же представлены знаки, идентифицирующие эти объекты.

Возникает вопрос, о каких различных реализациях множества идет речь. Разве множество не имеет раз и навсегда определенную реализацию? Например, какие другие реализации множества целых положительных чисел могут рассматриваться кроме традиционного – $\{1, 2, 3, 4, \dots\}$. В абстрактном мире математики подобное действительно не уместно. А вот при моделировании в БД динамичных ПрО одному и тому же

множеству в различные моменты времени могут соответствовать различные реализации. В качестве примера можно рассмотреть множество студентов, сидящих в конкретной аудитории. Его экстенционал будет меняться чуть ли не на каждой паре занятий при неизменном интенсionaле.

Оперируя понятиями множества и кортежа, можно формально определить совокупность объектов, специфицировав условия принадлежности и задав порядок. Однако такие совокупности не интерпретированы, так как они не соотнесены с объектами реального мира. Необходимо ввести правила придания множествам семантической окраски.

Определение 2.2.10. Существуют некоторые множества, элементы которых более или менее синтаксически однородны, например, множество целых чисел от 10 до 20, строки букв длиной до 20 символов и т.д. Подобные однородные множества в моделировании данных носят название **доменов**. Домены можно рассматривать как множества, из которых черпаются семантически значимые знаки для представления в знаковой системе объектов и значений их характеристик. Например, шестизначные числа образуют домен значений заработной платы. Слово «домен» в отличие от его употребления в математике обозначает как область определения, так и область значения отображений.

Следует отметить, что домены могут быть однородными как чисто синтаксически (все их элементы удовлетворяют одному и тому же синтаксису), так и семантически (в их условии принадлежности фигурируют особенности конкретной ПрО). Примерами доменов первого вида являются *домен целых положительных чисел (ЦПЧ)*, *домен строк символов компьютерного алфавита*, *домен дат* и т.п. Домены с однородностью второго типа являются специфическими подмножествами доменов первого типа и, естественно, предполагают также и синтаксическую однородность. Примерами таких доменов являются *домен номеров телефонов города Томска* (подмножество целых положительных чисел в диапазоне от 0 до 999999, определяемое телефонными пулами АТС города Томска), *домен дат рождения взрослого населения России*, *домен потенциально возможных значений характеристики Рост В Сантиметрах людей* и т.д.

Домены второго типа (с семантической однородностью) с одной стороны позволяют определять дополнительные ограничения целостности на уровне домена, с другой стороны могут способствовать тому, что при манипулировании данными будут исключены «неразумные» выражения, такие как сравнение значения характеристики *Рост* со значением характеристики *Телефон* или вычисление значения характеристики *Рост* путем арифметических действий, в которых участвует значение характеристики *Телефон*.

Определение 2.2.11. Именованные домены, представляющие семантически значимые объекты, называются **атрибутами**. Атрибуты являются интерпретацией объектов реального мира и их характеристик. Вводя атрибуты, мы даем интерпретацию абстрактным понятиям, таким, как числа и строки.

Домен можно рассматривать как обобщение атрибутов (смотрите иерархию обобщения в левой нижней части последнего слайда). Атрибуты, определенные на общем домене, наследуют его свойства. И, наоборот, домен обладает всеми общими свойствами определенных на нем атрибутов. Обобщение в применении к атрибутам позволяет соотнести структуру данных с доменами и установить общность различных атрибутов.

Атрибуты существуют не сами по себе, а как компоненты других объектов БД. Посредством агрегации они ассоциируются с другими атрибутами. Например, *Фамилия*, *Адрес* и *Возраст* формируют агрегат *ЛИЧНОСТЬ*. Интерпретация атрибутов и соотношений между ними определяется агрегатами, соответствующими понятиям реального мира. Прежде чем непосредственно обратиться к этим объектам, необходимо рассмотреть структуру, на которой они базируются. Эта структура может быть представлена множествами, элементы которых имеют свою внутреннюю структуру, т.е. отношениями.

Отношение

Математика

n -местным отношением R на множествах A_1, \dots, A_n называется подмножество прямого произведения $A_1 \times \dots \times A_n$. Другими словами, элементы x_1, \dots, x_n , где $x_i \in A_i (i = 1, \dots, n)$, связаны отношением R тогда и только тогда, когда $\langle x_1, \dots, x_n \rangle \in R$.

Пусть R – бинарное отношение. Определим обратное отношение R^{-1} следующим образом: $R^{-1} = \{\langle x, y \rangle \mid \langle y, x \rangle \in R\}$. Таким образом, R^{-1} связывает те же пары элементов, что и R , но «в другом порядке».

Логика

Пусть $E^n = E \times \dots \times E$ есть произведение n множеств E , т.е. множество всех кортежей $\langle x_1, \dots, x_n \rangle$, $x_i \in E (i = 1, \dots, n)$. Отображение $P: E^n \rightarrow \{0, 1\}$ называется n -местным отношением (предикатом, логической функцией) над E . Множество $A \subseteq E^n$ всех кортежей, для которых $P(x_1, \dots, x_n) = 1$, определяет «свойство» кортежей: x_1, \dots, x_n состоят в отношении P тогда и только тогда, когда $\langle x_1, \dots, x_n \rangle \in A$.

Рассмотрим ряд множеств, каждое из которых представляет некоторый тип. Множества могут интерпретироваться посредством доменов, ролей и атрибутов. Агрегация множеств позволяет сформулировать более сложные типы. Этому процессу можно придать формальный характер, абстрагируясь от интерпретации агрегации и выделяя общие свойства агрегатов. В таком случае агрегат, построенный на множествах, естественно представлять как отношение.

Определение 2.2.12. Математическое отношение – это множество, выражающее соответствие между двумя или более множествами. Соответствие между двумя множествами S_1 и S_2 называется бинарным отношением, если оно является подмножеством прямого (Декартова) произведения S_1 и S_2 , т.е. подмножеством $S_1 \times S_2$ ($S_1 \times S_2 = \{\langle s_1, s_2 \rangle \mid s_1 \in S_1 \wedge s_2 \in S_2\}$). Бинарное отношение задается множеством упорядоченных пар $\langle s_1, s_2 \rangle$, которые удовлетворяют некоторому критерию. Порядок элементов кортежа важен, что иллюстрируется следующим примером.

Рассмотрим два множества $S_1 = \{1, 3, 8, 9\}$ и $S_2 = \{2, 3\}$ и отношение $R_{<} \subseteq S_1 \times S_2$ такое, что $R_{<} = \{\langle s_1, s_2 \rangle \mid s_1 \in S_1 \wedge s_2 \in S_2 \wedge s_1 < s_2\}$. Прямое произведение этих множеств есть множество $\{\langle 1, 2 \rangle, \langle 3, 2 \rangle, \langle 8, 2 \rangle, \langle 9, 2 \rangle, \langle 1, 3 \rangle, \langle 3, 3 \rangle, \langle 8, 3 \rangle, \langle 9, 3 \rangle\}$, а отношение $R_{<}$ – его подмножество $\{\langle 1, 2 \rangle, \langle 1, 3 \rangle\}$. Заметим, что кортеж (а элементами отношения являются как раз кортежи) $\langle 1, 2 \rangle$ является элементом этого отношения, а $\langle 2, 1 \rangle$ – нет.

В общем случае отношение R может быть n -арным, т.е. представлять собой подмножество прямого (Декартова) произведения n множеств $R \subseteq S_1 \times S_2 \times \dots \times S_n$.

Определение 2.2.13. Так же как и любое другое множество, отношение может быть описано посредством интенционала и экстенционала. **Интенционал отношения** определяется интенционалами входящих в отношение множеств и общим свойством элементов собственно отношения. Для приведенного примера интенционал отношения $R_{<}$ задается выражением $R_{<} = \{\langle s_1, s_2 \rangle \mid s_1 \in S_1 \wedge s_2 \in S_2 \wedge s_1 < s_2\}$. **Экстенционал отношения** определяет одну из возможных его реализаций путем явного указания его кортежей. В нашем случае экстенционал отношения $R_{<}$ это $\{\langle 1, 2 \rangle, \langle 1, 3 \rangle\}$.

Мы познакомились с определением понятия «отношение», традиционно используемым в математике. К сожалению, есть другие определения этого понятия, применяемые в других областях знаний. Это естественно затрудняет восприятие человеком соответствующих теорий, т. к. вынуждает его переключаться с одной системы понятий на другую. Следующее определение, в частности, используется в логике.

Определение 2.2.14. Пусть $E^n = E \times \dots \times E$ есть произведение n множеств E , т.е. множество всех кортежей $\langle x_1, \dots, x_n \rangle$, $x_i \in E (i = 1, \dots, n)$. Отображение $P: E^n \rightarrow \{0, 1\}$

называется **n -местным отношением (предикатом, логической функцией) над E** . Множество $A \subseteq E^n$ всех кортежей, для которых $P(x_1, \dots, x_n) = 1$, определяет «свойство» кортежей: x_1, \dots, x_n состоят в отношении P тогда и только тогда, когда $\langle x_1, \dots, x_n \rangle \in A$.

**Отношение
Моделирование данных**

Пусть задано множество из n типов или доменов $T_i (i=1, \dots, n)$, причем все они необязательно должны быть различными. Тогда r будет отношением, определенным на этих типах, если оно состоит из двух частей: заголовка и тела (заголовок еще иногда называют схемой или интенционалом отношения, а тело – экстенционалом отношения), где:

- заголовок – это множество из n атрибутов вида $A_i : T_i$; здесь A_i – имена атрибутов отношения r , а T_i – соответствующие имена типов;
- тело – это множество из m кортежей t ; здесь t является множеством компонентов вида $A_i : v_i$, в которых v_i – значение типа T_i , т.е. значение атрибута A_i в кортеже t .

Наконец, в моделировании данных традиционно используется следующее определение понятия «отношения».

Определение 2.2.15. Пусть задано множество из n типов или доменов $T_i (i=1, \dots, n)$, причем все они необязательно должны быть различными. Тогда r будет **отношением**, определенным на этих типах, если оно состоит из двух частей: заголовка и тела (заголовок еще иногда называют схемой или интенционалом отношения, а тело – экстенционалом отношения), где:

- заголовок – это множество из n атрибутов вида $A_i : T_i$; здесь A_i – имена атрибутов отношения r , а T_i – соответствующие имена типов;

- тело – это множество из m кортежей t ; здесь t является множеством компонентов вида $A_i : v_i$, в которых v_i – значение типа T_i , т.е. значение атрибута A_i в кортеже t .

Попытаемся сопоставить все приведенные ранее определения. Человека, прежде всего, интересует истинность или ложность того или иного факта, поэтому ему ближе и понятней определение логики (2.2.14). В информационных системах (и в базах данных, в частности) принято хранить только истинные факты и предполагать выполнение гипотезы о замкнутости мира: если кортеж не содержится в теле отношения, то соответствующее ему утверждение ложно. Понятно, что в таком случае первое (2.2.12) и второе (2.2.14) определения эквивалентны. Третье определение (2.2.15) отличается от первого (2.2.12) отсутствием порядка в элементах кортежей, и, таким образом, одно отношение в смысле третьего определения (2.2.15) соответствует сразу нескольким отношениям, в смысле первого определения (2.2.12). Именно поэтому третье определение (2.2.15) используется в базах данных.

Отношение			
Брак		Брак ⁻¹	
Муж	Жена	Жена	Муж
Джон	Мэри	Мэри	Джон
Джек	Сара	Сара	Джек
Джим	Лаура	Лаура	Джим

Заголовок – {Муж: Строки, Жена: Строки}

A_1	T_1	A_2	T_2
-------	-------	-------	-------

Тело – {

{Муж: Джон, Жена: Мэри},	t_1
{Муж: Джек, Жена: Сара},	t_2
{Муж: Джим, Жена: Лаура}	t_3

}

A_1	v_1	A_2	v_2
-------	-------	-------	-------

На слайде приведено отношение *БРАК* в традиционной табличной форме представления (подробнее о ней речь пойдет чуть далее). Если использовать математическое определение понятия «отношение», то обратное ему отношение *БРАК*⁻¹ будет представляться таблицей, в которой изменен порядок столбцов (смотрите слайд).

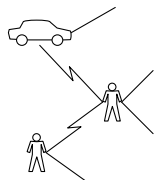
Мы же будем придерживаться определения 2.2.15, в котором порядок столбцов и строк в таблице, представляющей отношение, не играет никакой роли. Действительно, не дураки же мы, чтобы хранить в БД и отношение *БРАК*, и отношение *БРАК*⁻¹. Это для бинарных отношений их всего два. В случае отношения степени n количество различных вариантов перестановок столбцов равно $n!$. Это ж сколько понадобится лишней памяти и времени для синхронной модификации всех этих отношений. Поэтому в моделировании данных в качестве формы используют отношение именно в смысле последнего определения.

В нижней части слайда на примере того же отношения *БРАК* поясняются элементы, используемые в этом определении.

Продолжим сопоставление отношений в смысле первого и третьего определений. В математике отношение – это не более чем множество, и оно не имеет какой-либо семантической интерпретации. В моделировании данных термин «отношение» применяется к определению агрегированного типа. Отношение R рассматривается как тип объектов, который соотносится с множеством знаков-кортежей, представляющих сами объекты. Все кортежи определены на одних и тех же доменах и обладают одинаковыми признаками.

Следующее отличие аналогично уже упоминавшемуся отличию между множествами в математике и в моделировании данных. В силу динамичности ПрО одному и тому же интенсionalу отношения в разные моменты времени могут соответствовать различные экстенсionalы. Действительно, коль скоро кортежи отношения представляют объекты ПрО, также как и последние, кортежи отношения могут появляться, исчезать, а также изменять свои элементы.

И, наконец, если в математике условие, при истинности которого кортежи Декартова произведения попадают в отношение, может быть выражено формально (условие $s_1 < s_2$ в примере с отношением $R_<$), то в моделировании данных это условие не может быть представлено формально. Более того, эту задачу никогда не удастся автоматизировать, поскольку она является прерогативой человека. Адекватное изменениям ПрО преобразование экстенсionalов отношений БД – главная задача ранее упоминавшихся пользователей-писателей.

Типичное использование форм данных		
Предметная область	Модель человека	Модель данных
	Объект	Кортеж отношения
	Тип объектов	Отношение
	Связь	Кортеж отношения
	Тип связей	Отношение
	Характеристика	Атрибут отношения
	Значение характеристики	Значение атрибута в кортеже

Памятуя об основной функции любой модели данных – обеспечении интерпретации данных, переходим к рассмотрению традиционных и наиболее естественных способов интерпретации данных, представленных в приведенных ранее формах.

Основной принцип очень прост – интерпретации наследуются в соответствии с иерархией обобщения. Действительно, поскольку каждый знак (данное) состоит в отношении «есть некоторый» со своим типом, для него, естественно, истинно условие принадлежности (интенционал) множества, в форме которого представлены все знаки этого типа.

В схеме БД интенционалы множеств, как правило, представлены именами типов, которые можно трактовать как предикаторы (имена) одноместных предикатов-свойств. Например, с типом данных *СТУДЕНТ* можно ассоциировать одноименный предикат *СТУДЕНТ(x)*, который принимает значение *Истина*, если объект, обозначаемый знаком *x* в действительности является студентом.

Поскольку, как и объекты ПрО, знаки и типы вступают между собой во взаимосвязи не только в соответствии с отношением обобщения, нам не обойтись без сложных знаков-агрегатов и типов-агрегатов. Механизм интерпретации и в этом случае работает по тому же принципу.

Для уточнения картины проведем соответствие между понятиями, используемыми людьми в процессе мировосприятия, и формами, в которых представляются данные в моделях данных. Объект, обладающий совокупностью индивидуальных значений характеристик, естественно представлять в виде кортежа этих значений. Для всех объектов одного типа выделяется одинаковый набор характеристик, который и составляет множество атрибутов отношения, представляющего этот тип объектов. Следует отметить, что один и тот же объект в контексте разных понятий (типов) может представляться кортежами разных отношений. Дубликаты кортежей в одном отношении, естественно, не уместны.

Интерпретация отношений как типов объектов не является единственной. Другая возможная их интерпретация – тип связей между типами объектов. Тип связей представляет собой агрегат двух или более типов объектов. Естественно, что каждая связь конкретного типа задается в БД в виде кортежа, элементами которого являются знаки соответствующих объектов.

И, наконец, связи, как и объекты, также могут описываться значениями характеристик, которые агрегируются с кортежами-связями. На уровне типов это влечет за собой добавление атрибутов в отношение, соответствующее типу связей.

Представление данных в виде таблиц					
СЛУЖАЩИЙ				КОМПАНИЯ	
№ СЛУЖАЩЕГО	ФАМИЛИЯ	АДРЕС	ПОЛ	НАЗВАНИЕ	ГОРОД
123	СМИТ	ЭВЕЛИН, 19	Ж	АЕС	ТОРОНТО
862	ЛОКК	ШАТЕР, 85	М	ИБМ	НЬЮ-ЙОРК
781	БЭРР	КУИН, 16	М		
523	ДЖОНС	ДЖОРДЖ, 3	Ж		
324	ПИТЦ	ЛАНДИ, 22	Ж		
СЛУЖБА					
ВАРИАНТ 1			ВАРИАНТ 2		
№ СЛУЖАЩЕГО	ФАМИЛИЯ	АДРЕС	ПОЛ	НАЗВАНИЕ	ГОРОД
123	СМИТ	ЭВЕЛИН, 19	Ж	АЕС	ТОРОНТО
862	ЛОКК	ШАТЕР, 85	М	АЕС	ТОРОНТО
781	БЭРР	КУИН, 16	М	АЕС	ТОРОНТО
523	ДЖОНС	ДЖОРДЖ, 3	Ж	ИБМ	НЬЮ-ЙОРК
123	СМИТ	ЭВЕЛИН, 19	Ж	ИБМ	НЬЮ-ЙОРК
ВАРИАНТ 2		НАЗВАНИЕ		ГОРОД	
№ СЛУЖАЩЕГО	НАЗВАНИЕ	№ СЛУЖАЩЕГО	НАЗВАНИЕ	№ СЛУЖАЩЕГО	НАЗВАНИЕ
123	АЕС	123	АЕС	123	АЕС
862	АЕС	862	АЕС	862	АЕС
781	АЕС	781	АЕС	781	АЕС
523	ИБМ	523	ИБМ	523	ИБМ
123	ИБМ	123	ИБМ	123	ИБМ

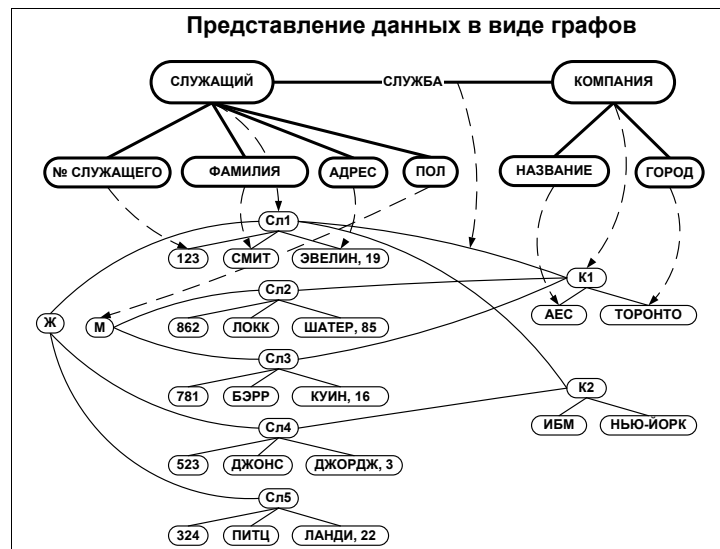
Введенные ранее формы данных носят абстрактный характер и не имеют какого-либо предопределенного представления. Между тем в моделировании представление является весьма важным вопросом, поскольку необходимо, во-первых, хранить данные в памяти ЭВМ, во-вторых, представлять их пользователю в форме, облегчающей интерпретацию и понимание.

Возможно, простейшим и широко распространенным способом представления данных и их интерпретаций являются **таблицы**. С ними люди привыкли работать повседневно. Табличные формы представления характерны и для большинства моделей данных.

Объекты одного типа представляются отдельной таблицей (*СЛУЖАЩИЙ* и *КОМПАНИЯ*), олицетворяющей соответствующее отношение. Заголовок таблицы (отношения) представляет собой общее имя и характеризует соответствующее понятие о предметах. Каждая строка таблицы (кортеж) представляет конкретный объект и задает его значения однозначных характеристик, общих для всех объектов одного типа. Каждый столбец таблицы представляет один атрибут, имя которого помещается в шапку таблицы.

Агрегаты типов объектов – типы связей – также могут быть представлены таблицами, олицетворяющими соответствующие отношения. В табличной форме связи задаются дублированием значений атрибутов соответствующих объектов. На рисунке представлены два варианта для таблицы связей типа *СЛУЖБА*. В первом случае используется полное дублирование значений всех атрибутов объектов. Это представление избыточно, так как связи между объектами однозначно определяются указанием значений их ключевых атрибутов. Ключевой атрибут – это атрибут, уникально идентифицирующий объект и являющийся его знаком. Такими в нашем случае являются атрибуты *№ Служащего* для служащих и *Название* для компаний. Только эти атрибуты и представлены в таблице *СЛУЖБА* во втором случае.

Обратите внимание, мы вынуждены дублировать данные об объекте в таблице связей в том случае, когда он имеет несколько связей одного типа (в нашем случае служащий с номером 123 работает одновременно в двух компаниях).



Многие модели данных предлагают представлять схему и БД в виде **графов**. Граф схемы часто называют **графом типов**, поскольку в нем представлены именно типы, а граф БД в таком случае носит название **графа знаков**, так как вершинами этого графа являются знаки. Граф типов для той же ПрО, что и в примере с таблицами, показан в верхней части рисунка, граф знаков – в нижней. Несмотря на то, что это различные графы, между вершинами и ребрами этих графов по умолчанию предполагается наличие отношений экземпляризации, часть которых показана на рисунке в виде пунктирных дуг.

Именованные вершины графа типов представляют атрибуты (*№ Служащего, Фамилия, Адрес, Пол, Название, Город*) и типы объектов (*СЛУЖАЩИЙ, КОМПАНИЯ*). Совокупность не имеющих имен ребер графа типов, соединяющих атрибуты с их типом объектов, соответствует отношению агрегации типов. Именованные ребра этого графа олицетворяют типы связей (*СЛУЖБА*) – агрегаты типов объектов.

Вершины графа знаков представляют объекты (*Сл1, Сл2, Сл3, Сл4, Сл5, К1, К2*) и значения (все остальные вершины). Совокупность ребер графа знаков, соединяющих значения с их объектом, соответствует отношению агрегации знаков. Остальные ребра графа знаков, на концах которых расположены вершины-объекты, олицетворяют связи между этими объектами. В нашем примере связи типа *СЛУЖБА* бинарны, поэтому у каждого ребра таких связей 2 конца. В случае связей степени n у этих ребер будет n концов.

Характерной особенностью графового представления является отсутствие каких либо дубликатов (обратите внимание на единственность значений атрибута *Пол* «М» и «Ж», а также служащего с номером 123, представленного одной вершиной «Сл1»).

Во многих моделях данных предусматривается одновременное использование графов и таблиц. Наиболее широко распространенная форма подобного «симбиоза» – граф с вершинами-таблицами. Хотя иногда графы и таблицы предлагаются как независимые альтернативы представлений данных и их интерпретаций.

Вопросы и задания к параграфу 2.2

1. Для чего в технологии БД используются знаки и типы?
2. В каких частях системы БД представлены знаки и типы?
3. Какие мыслительные процедуры используются при структуризации данных?
Уточните, когда применяется каждая из них.
4. Поясните, как абстракции используются для интерпретации данных (знаков).
5. Перечислите традиционно используемые в моделях данных формы данных.
6. Укажите свойства множеств.

7. Чем отличается комплекс от множества?
8. Покажите, что понятия «множество» и «кортеж» являются специализациями понятия «комплекс».
9. Дайте определения интенционала и экстенционала множества.
10. Почему в моделировании данных уместно говорить о нескольких реализациях множества?
11. В чем принципиальное отличие элементов, принадлежащих домену и атрибуту?
12. Уточните различия определений понятия «отношение» в математике, логике и моделировании данных.
13. В каком виде задаются в БД интерпретации данных?
14. Каким формам данных приписываются интерпретации?
15. Как они используются для интерпретации знаков (данных)?
16. Установите соответствие между формами данных и их представлениями в виде таблиц и графов.