

ГЛАВА 2.1

Вопросы и задания к параграфу 2.1

1. Объясните, почему человека интересуют не данные, а информация.
2. Что кроме данных необходимо для получения информации?
3. Перечислите и охарактеризуйте три этапа процесса образования информации из данных.
4. Какие разделы семиотики изучают эти этапы?
5. К какому классу информационных систем относятся системы БД?
6. Опишите, как протекает процесс передачи информации о ПрО с использованием систем БД.
7. В чем заключается основное назначение модели данных?
8. Из каких компонентов состоит атомарная единица информации (АЕИ)?
9. Объясните, почему первые три компонента АЕИ являются обязательными.
10. С чем ассоциируются понятия «схема БД» и «база данных» при табличном представлении данных?
11. Дайте структурное определение модели данных.
12. Проиллюстрируйте компоненты модели данных на примере категориальной модели.
13. Что представляет собой СУБД?

1.

Стремление к приобретению знаний присуще самой природе человека. Наиболее эффективный способ получения новых знаний – проведение научных исследований. Вместе с тем любой индивидуум, наблюдая некоторое явление, получает определенное приращение знаний, которое мы будем называть информацией. Информация может обладать большой ценностью и подлежать регистрации для обеспечения ее доступности множеству заинтересованных лиц. Достижение этой цели существенно связано с представлением информации.

2.

Изначальным средством представления информации служит естественный язык. Однако это не всегда наилучшее средство. Во-первых, естественный язык в силу своей универсальности в некоторых случаях менее эффективен, чем специализированные средства. Во-вторых, естественный язык ориентирован на вольный стиль общения и не обеспечивает необходимой точности регистрации и передачи информации. И, наконец, в силу свойств естественного языка обработка созданных с его помощью текстов на компьютере связана с принципиальными трудностями.

Наряду с решением сложных вычислительных задач компьютеры используются как интеллектуальные помощники, обеспечивающие управление и манипулирование информацией. Вероятно, это останется главной функцией компьютера и в перспективе. Эффективность ее реализации зависит от способов компьютерного представления данных и информации.

3.

4.

Определение 2.1.4. Синтактика – раздел семиотики, изучающий внутренние свойства систем знаков безотносительно к интерпретации (синтаксис – правила построения знаков и знакосочетаний в рамках знаковой системы).

Определение 2.1.5. Семантика – раздел семиотики, рассматривающий отношение знаков к обозначаемому (содержание знаков) или, что то же, соотношения между знаками и их интерпретациями, независимо от того, кто служит «адресатом» (интерпретатором).

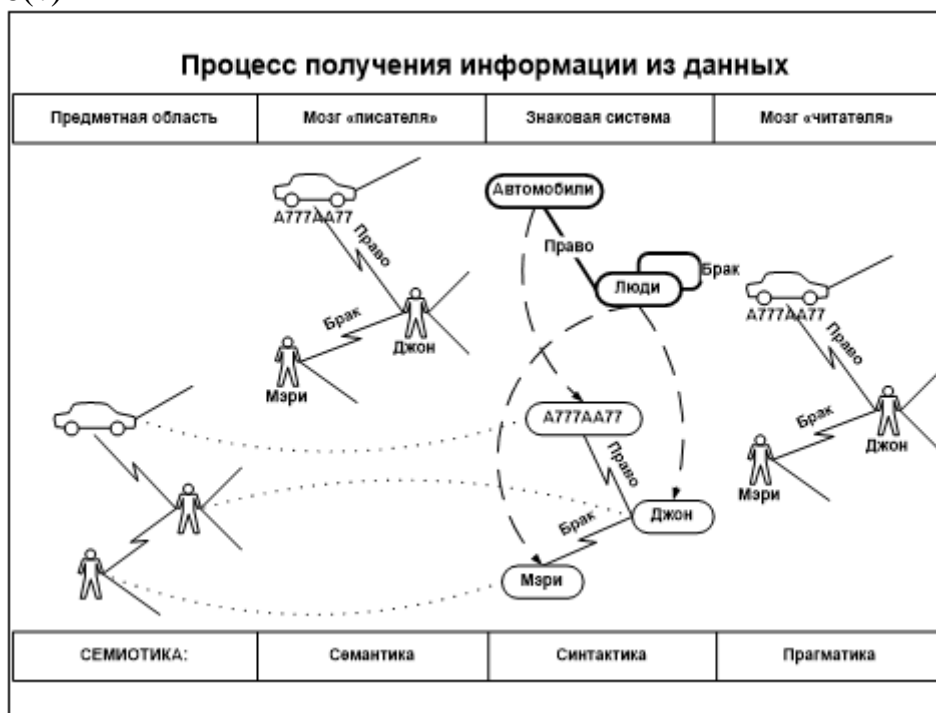
Термин «семантически значимый», который мы в дальнейшем будем часто использовать, предполагает значимость с точки зрения семантики, изучающей семантические отношения, которые образуются между объектами и знаками, представляющими эти объекты в знаковой системе. Таким образом, семантически значимыми могут быть как объекты (если они представлены некоторыми знаками в семиотической системе), так и знаки (если они определяют некоторые реальные объекты ПрО).

Определение 2.1.6. Прагматика – раздел семиотики, изучающий восприятие выражений знаковой системы в соответствии с разрешающими способностями воспринимающего. Прагматика исследует связь знаков с «адресатом», т. е. проблемы интерпретации знаков теми, кто их использует, их полезность и ценность для интерпретатора.

5(?)

По такому принципу строили ИС в доСУБДэшную эпоху – управление данными в файле осуществлялось специализированными программами (для каждого файла свои программы). На смену им пришли СУБД – универсальные программные комплексы, предназначенные для управления БД о любых ПрО. Это стало возможным благодаря тому, что в них интерпретации не фиксируются в программах, а хранятся на диске, как и сами данные. Именно их мы задаем на этапе проектирования схемы БД. Заданные однократно интерпретации ассоциируются с данными при их вводе, и в дальнейшем при всех манипуляциях с данными эта связь остается неразрывной, обеспечивая, таким образом, получение информации.

6(?)



Определение 2.1.7. (Функциональное определение модели данных). Модель данных (МД) – это интеллектуальное средство, позволяющее реализовать интерпретацию данных и таким образом способствующее получению информации. Громкий эпитет «интеллектуальное» применен в данном случае неспроста – способность к восприятию, хранению и передаче информации – одно из важнейших свойств интеллекта.

7

Как вы уже поняли, основными объектами исследований в моделировании данных являются данные и их интерпретации. Эта область человеческих знаний, как и многие другие, имеет свой элементарный объект – **атомарную единицу информации (АЕИ)**. Она определяется четверкой – *<Идентификатор объекта, Наименование признака, Значение признака, [Время]>*.

Последний элемент четверки не напрасно заключен в квадратные скобки, означающие, что он может быть опущен. Действительно, все остальные элементы являются обязательными. Если опустить хотя бы один из них, информация не образуется. Что же касается времени, то не существует моделей данных, в которых механизмы работы с так называемыми темпоральными данными были бы доведены до идеала. Более того, подавляющее большинство моделей и БД предполагают хранение информации об одном (текущем) состоянии ПрО. Такие БД еще называют оперативными, подчеркивая тот факт, что они меняются синхронно с изменением состояния ПрО. Предыдущие состояния данных в них не сохраняются. Поэтому в дальнейшем под АЕИ будем понимать тройку – *<Идентификатор объекта, Наименование признака, Значение признака>*. Примеры АЕИ приведены на слайде.



Определение 2.1.8. Совокупность именованных категорий и их признаков, а также ограничений на допустимые данные называется **схемой БД**.

Определение 2.1.9. Совокупность данных, структура и значения которых соответствуют конкретной схеме, называется **базой данных (БД)**.

По поводу этих определений следует сделать несколько замечаний.

Во-первых, как мы увидим в дальнейшем, каждая модель данных предполагает свой набор понятий, используя которые мы структурируем Про. Наши определения даны для категориальной модели, поэтому в них используются понятия «категория» и «признак». Тем не менее, лучше использовать конкретные, но конструктивные определения, чем давать «размытые» определения типа «база данных – это некоторый набор перманентных данных, используемых прикладными системами какого-либо предприятия».

Во-вторых, в этих определениях указано основное содержание понятий «схема» и «база данных». Иногда это содержание так или иначе расширяют. Так в схему БД могут быть включены программы обработки данных, определения запросов, диалоговых форм и т.д. БД тоже может представляться как все, что хранится на диске помимо собственно СУБД. Мы и сами иногда будем поступать так. В частности, на приведенной схеме БД вобрала в себя помимо собственно данных еще и их схему.

11

Определение 2.1.10. (Структурное определение модели данных). Модель данных (МД) определяется двумя множествами G и O . G – множество правил порождения схем, O – множество операций над данными. В свою очередь во множестве G выделяются два подмножества – G_s (правила порождения структур данных) и G_c (правила порождения ограничений целостности).

Поясним указанные компоненты моделей данных. Для нашей категориальной модели правила множества G_s могут выглядеть следующим образом:

- БД – это совокупность таблиц.
- Каждая таблица предназначена для хранения информации об объектах одной категории. Имя таблицы – это имя категории.
- Для каждой категории определяется набор признаков, представляющих интерес для объектов этой категории. Имена признаков составляют шапку соответствующей таблицы.
- Каждый объект категории представляется в виде строки таблицы, в столбце признака указывается его значение для данного объекта.

Мы пока детально не обсуждали ограничения целостности, накладывающие дополнительные условия на вводимые в БД данные, тем не менее, приведем несколько правил множества G_c .

Допустимые значения признаков можно ограничить:

- указанием их типа (символьные, числовые, даты и т.д.),
- перечислением этих значений,
- сравнением значений с константой.

Множество операций над данными O может выглядеть для нашей модели так:

- операция *INSERT* для добавления новой строки в таблицу,
- операция *UPDATE* для изменения значений одного или нескольких признаков в строке таблицы,
- операция *DELETE* для удаления строки из таблицы,
- операция *SELECT* для поиска строк таблицы, удовлетворяющих определенному условию.

Определение 2.1.11. Управление БД на ЭВМ осуществляется специализированными программными средствами – **системами управления базами данных (СУБД)**, каждая из которых предлагает свои языковые и диалоговые формы для множеств G и O : язык определения данных (ЯОД) и язык манипулирования данными (ЯМД). Иногда выделяют отдельный язык определения ограничений целостности (ЯООЦ), но чаще ограничения целостности задаются вместе со структурой в командах ЯОД.

Все приведенные здесь основные понятия технологии БД тесно взаимосвязаны и образуют целостную систему, представленную на последнем слайде. Любое моделирование данных невозможно без использования той или иной модели (левый

24

столбец). Для этого в ней должны быть предусмотрены множества G_s , G_c и O . Для создания реальных БД в ней модель должна поддерживаться СУБД (средний столбец), в которой составляющие модель множества превращаются в соответствующие языковые средства (ЯОД, ЯООЦ и ЯМД). Передавая СУБД команды этих языков, человек создает схему БД (структуры и ограничения целостности), а затем и сами данные (правый столбец). Дерево, приведенное на слайде справа, иллюстрирует взаимосвязи понятий «модель данных» (M), «схема БД» (S_i) и «состояние БД» (D_i).

13

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.

41

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.

1 - 2

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.

3 - 4(?)

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.

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.

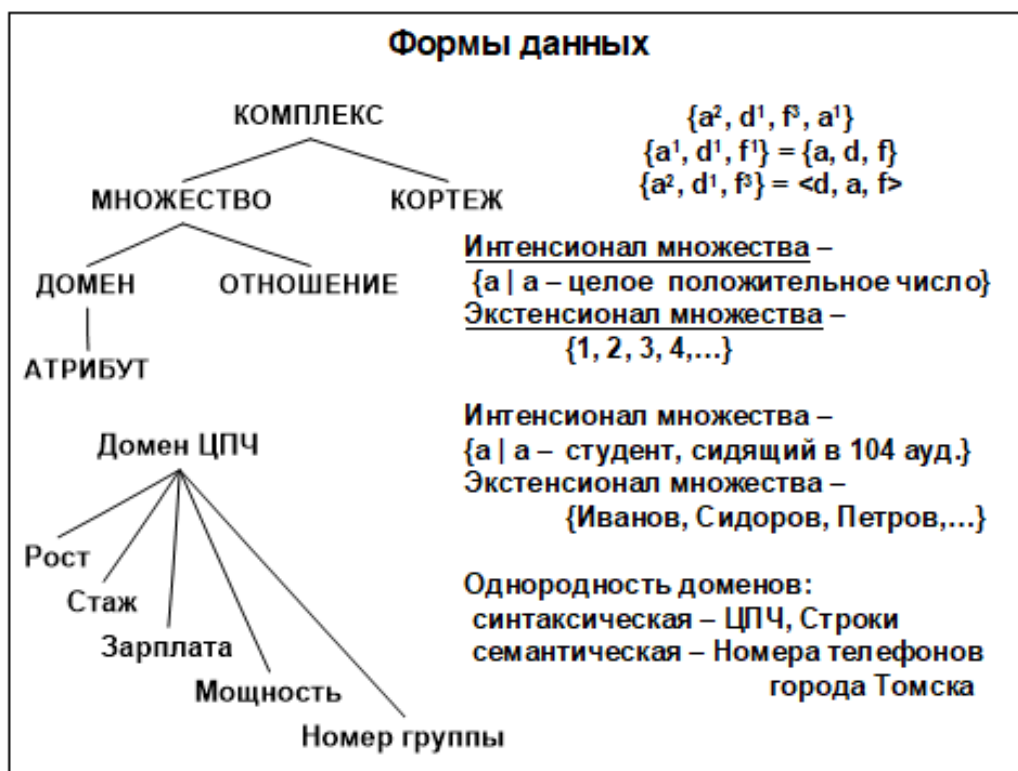
Определение 2.2.5. Агрегация – это абстракция, посредством которой объект конструируется из других, базовых объектов в соответствии с отношением «есть часть» (англ. *part of*). Агрегация возможна как на уровне знаков (сложный знак собирается из более простых знаков), так и на уровне типов (сложный тип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

27

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 экземплиризации, специализации и декомпозиции.

A



6

В классической теории множеств эта форма характеризуется следующими свойствами:

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

7

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.

8

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.

9

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.

10

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

33

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 интенционале.

11(???)

Атрибут является производной домена и т.д.

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.

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.

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 2.2.14. 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 $P : E^n \rightarrow \{0, 1\}$

35

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 $\langle x_1, \dots, x_n \rangle \in A$.

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 .

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 $\langle x_1, \dots, x_n \rangle \in A$.

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 (заголовков 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 .

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.

II, 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.

Вопросы и задания к параграфу 2.3

1. Что такое ограничение целостности (ОЦ) и для чего они предназначены?
2. Охарактеризуйте способы задания ОЦ с точки зрения их предпочтительности.
3. Какие типы ОЦ выделяются в моделировании данных? Каковы их области действия?
4. Перечислите виды ОЦ на значения атрибутов.
5. В чем особенность традиционного определения понятия «отображение» в математике?
6. Укажите основные отличия «математического отображения» от «семантически значимого отображения».
7. Почему в моделировании данных приходится рассматривать реальные и «потенциальные» ООО и ОЗО?
8. Поясните фразу «бинарное отношение множеств определяет два отображения между ними».
9. Дайте определения КЧ, МинКЧ и МаксКЧ. К каким элементам относятся эти характеристики?
10. Перечислите и укажите характеристики введенных типов отображений.
11. Какими символическими нотациями можно описать бинарные отношения? Сравните их информативность.
12. Укажите отличие сложного отображения от простого отображения.
13. Дайте три определения возможного ключа.
14. Что означает «определенность значения атрибута»?
15. Какие особенности каких отображений скрываются в определении возможного ключа и обязательности значений атрибута?
16. Что ассоциируется с понятиями «интенционал БД» и «экстенционал БД»?

1-2

Определение 2.3.1. Ограничение целостности (ОЦ) (англ. integrity constraint) можно представлять себе как логическое условие, которое для реализации данного множества, атрибута, отношения или нескольких отношений либо истинно, либо ложно.

ОЦ не всегда задаются в виде явного логического выражения, синтаксически они могут представлять собой просто декларацию определенных фактов. Однако механизм проверки ОЦ СУБД всегда использует их именно как логические условия, поскольку его задача – обеспечивать, чтобы для любого состояния БД все ОЦ, декларированные в ее схеме, были истинными.

В моделировании данных особую значимость имеют обобщенные ОЦ, т.е. ограничения, относящиеся к интенциональным свойствам данных, а не к свойствам данных конкретной реализации. Это значит, что ОЦ фиксируют в схеме общие закономерности данных о Про, которые выполнялись в прошлом, наблюдаются сейчас и будут справедливы и в будущем.

ОЦ вводятся в модели данных в целях повышения ее семантической и расширения возможностей поддержания целостности данных. Первый аспект связан с адекватностью отражения реального мира в схеме, а второй – с возможностями СУБД обеспечивать соответствие порождаемых состояний БД требованиям, выражаемым ОЦ.

Иногда выделяют внутренние и явные ОЦ. Первые тесно связаны с правилами структуризации, предполагаются самими формами данных и не требуют никаких дополнительных деклараций. Например, отсутствие дубликатов во множестве или отношении. Диапазон возможностей внутренних ограничений весьма узок. Поэтому в моделях данных предусматривают правила задания явных ОЦ, существенно расширяющие возможности передачи в схеме БД закономерностей Про. Для этого же в языке СУБД существуют специальные команды или опции команд для задания явных ОЦ. Чем больше ограничен диапазон структур данных, на представление которых ориентирована модель, тем большее число видов внутренних ОЦ она предусматривает и 3 тем меньшее число видов явных ОЦ необходимо и возможно задавать.

1) Ограничения целостности на значения атрибутов

Множество допустимых значений атрибутов можно задавать:

- 1) принадлежностью к определенному типу или домену
Рост INTEGER
- 2) сравнением с константой или значением атрибута того же отношения (возможны более сложные выражения)
Рост > 50 или Рост > Вес + 110
- 3) диапазоном
Рост BETWEEN 50 AND 300
- 4) перечислением значений
Пол IN {'м', 'ж'}
- 5) более сложным логическим выражением, включающим в виде атомов конструкции 2 – 4
(Пол = 'ж' AND Рост > Вес + 110) OR
(Пол = 'м' AND Рост > 180)

Прежде чем переходить к обсуждению ОЦ второго типа, необходимо познакомиться с самим понятием «отображение», являющимся одним из самых важных в моделировании данных и, тем не менее, часто передаваемым забвению. Это понятие так же, как и понятие «отношение», имеет различные трактовки, используемые в различных областях знаний. Как и ранее, обратимся к математике, в частности, к теории множеств.

В теории множеств понятие функции, геометрическое понятие отображения или преобразования фигуры и т. п. объединяются в общее понятие отображения одного множества в другое. Пусть даны два множества X и Y , пусть каждому элементу $x \in X$ поставлен в соответствие некоторый определенный элемент $y = f(x)$ множества Y ; тогда говорят, что имеется отображение множества X во множество Y , или, что имеется функция, аргумент x которой пробегает множество X , а значения y принадлежат множеству Y ; при этом для каждого данного $x \in X$ элемент $y = f(x)$ множества Y называется образом элемента $x \in X$ при данном отображении или значением данной функции для данного значения ее аргумента x .

Обратим ваше внимание на тот факт, что традиции, сложившиеся в математике, требуют обязательного наличия единственного образа для каждого аргумента. Правда даже в математике существуют отступления от этого правила, в частности, при рассмотрении многозначных функций.

Более полная картина разновидностей соответствий между элементами множеств наблюдается в теории нечетких множеств. У Кофмана самый общий случай так и называется соответствием. При этом не накладывается никаких ограничений – у конкретного аргумента (прообраза) образа может не быть вовсе или образов может быть произвольное число. Специализацией соответствия является отображение, требующее наличия хотя бы одного образа у каждого прообраза. И, наконец, специализацией отображения является функция, определение которой в точности соответствует первому

5 определению отображения.

Обратим ваше внимание на тот факт, что традиции, сложившиеся в математике, требуют обязательного наличия единственного образа для каждого аргумента. Правда даже в математике существуют отступления от этого правила, в частности, при рассмотрении многозначных функций.

Более полная картина разновидностей соответствий между элементами множеств наблюдается в теории нечетких множеств. У Кофмана самый общий случай так и называется соответствием. При этом не накладывается никаких ограничений – у конкретного аргумента (прообраза) образа может не быть вовсе или образов может быть произвольное число. Специализацией соответствия является отображение, требующее наличия хотя бы одного образа у каждого прообраза. И, наконец, специализацией отображения является функция, определение которой в точности соответствует первому определению отображения.

Определение 2.3.3. Семантически значимое отображение (англ. mapping) (в дальнейшем просто – отображение) – это понятие, определяющее некоторый закон предметной области, по которому каждому объекту моделируемого мира может быть поставлен в соответствие (а может быть, и нет) один или более объектов. Законы, действующие в мире объектов, представляются в знаковой системе в виде отображений, прообразами и образами которых являются знаки-объекты, знаки-связи и знаки-значения. Термин «семантически значимый» предполагает значимость с точки зрения семантики, изучающей семантические отношения, которые образуются между объектами и знаками, представляющими эти объекты в знаковой системе. Благодаря этому эпитету мы дистанцируемся от абстрактных (в частности, математических) отображений и функций.

7

Совокупность всех объектов, имеющих образы при отображении ϕ назовем реальной областью определения отображения (РООО) ϕ , а совокупность всех ϕ -образов этих объектов – реальной областью значений отображения (РОЗО) ϕ . Одним из самых главных отличий отображений объектов реального мира от функций в математике является динамичный характер экземпляров отображений. В течение времени эти экземпляры могут возникать и исчезать, у них могут меняться образы, в то время как функция $y = x^2$ раз и навсегда фиксирует свои экземпляры.

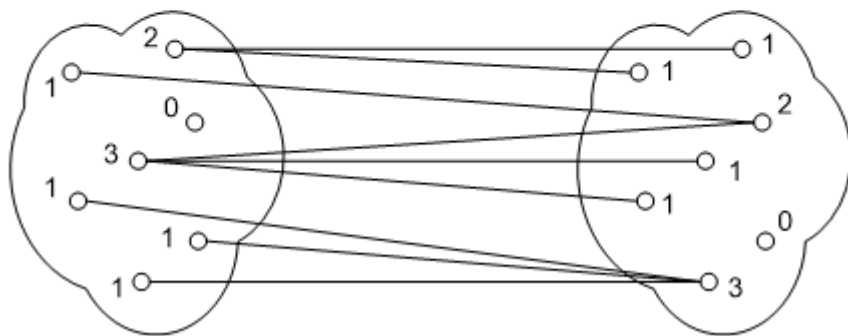
Семантика же отображений стабильна и в любой момент времени отражает тот закон, который справедлив для всех возможных состояний их экземпляров. По этой причине наряду с динамичными РООО и РОЗО для отображений необходимо рассматривать «потенциальные» области определения и значений отображений. **Область определения отображения ϕ (ООО)** (англ. domain) – это совокупность объектов, которые в силу смысла закона ϕ могли иметь в прошлом или имеют в настоящем или смогут иметь в будущем хотя бы один ϕ -образ. **Область значений отображения ϕ (ОЗО)** (англ. codomain) – это совокупность объектов, которые в силу смысла закона ϕ могли быть в прошлом или являются в настоящем или смогут стать в будущем ϕ -образами некоторых объектов. Например, если отображению *ДОЧЬ* придать юридический смысл, то ООО *ДОЧЬ* – совокупность всех людей, а ОЗО *ДОЧЬ* – совокупность всех женщин (*ДОЧЬ: ЧЕЛОВЕК -> ЖЕНЩИНА*).

Ниже представлены экземпляры обратного отображению *РЕБЕНОК* отображения *РОДИТЕЛЬ*. Очевидно, что они передают ту же самую информацию, только сменилась ориентация отображения, и объекты поменялись ролями (образы стали прообразами и наоборот). Это связано с тем, что отображение *РЕБЕНОК* эквивалентно инверсии отображения *РОДИТЕЛЬ* (об операциях над отображениями и эквивалентных отображениях речь пойдет в третьей главе), и оба этих отображения полностью определяются бинарным отношением *РОДИТЕЛЬ-РЕБЕНОК*, представленным в табличной форме в верхней части слайда. Если один из его атрибутов использовать как РООО, а второй – как РОЗО, каждый кортеж будет определять один экземпляр одного отображения. Если поменять атрибуты местами, те же кортежи определят экземпляры 8 обратного отображения.

Количественные характеристики отображений

φ -отображение объекта x характеризует **кардинальное число объекта x при отображении φ** (обозначение - $K\varphi(x)$) - мощность его области образов при этом отображении

$$\forall x \in \text{ЧЕЛОВЕК} : K\varphi_{\text{родитель}}(x) = 2$$



**Количественные характеристики отображений
(продолжение)**

Минимальное кардинальное число (МинКЧ) отображения φ - это наименьшее из кардинальных чисел $K\mathcal{C}_{\varphi}(x)$ объектов x , являющихся экземплярами области определения отображения $(ООО)_{\varphi}$:

$$MinK\mathcal{C}_{\varphi} = \min_{x \in \text{экземпляры}(ООО(\varphi))} K\mathcal{C}_{\varphi}(x)$$

$$MinK\mathcal{C}_{\text{родитель}} = 2 \quad MinK\mathcal{C}_{\text{ребенок}} = 0$$

Максимальное кардинальное число (МаксКЧ) отображения φ - это наибольшее из кардинальных чисел $K\mathcal{C}_{\varphi}(x)$ объектов x , являющихся экземплярами области определения отображения $(ООО)_{\varphi}$:

$$MaxK\mathcal{C}_{\varphi} = \max_{x \in \text{экземпляры}(ООО(\varphi))} K\mathcal{C}_{\varphi}(x)$$

$$MaxK\mathcal{C}_{\text{родитель}} = 2 \quad MaxK\mathcal{C}_{\text{ребенок}} = \infty$$

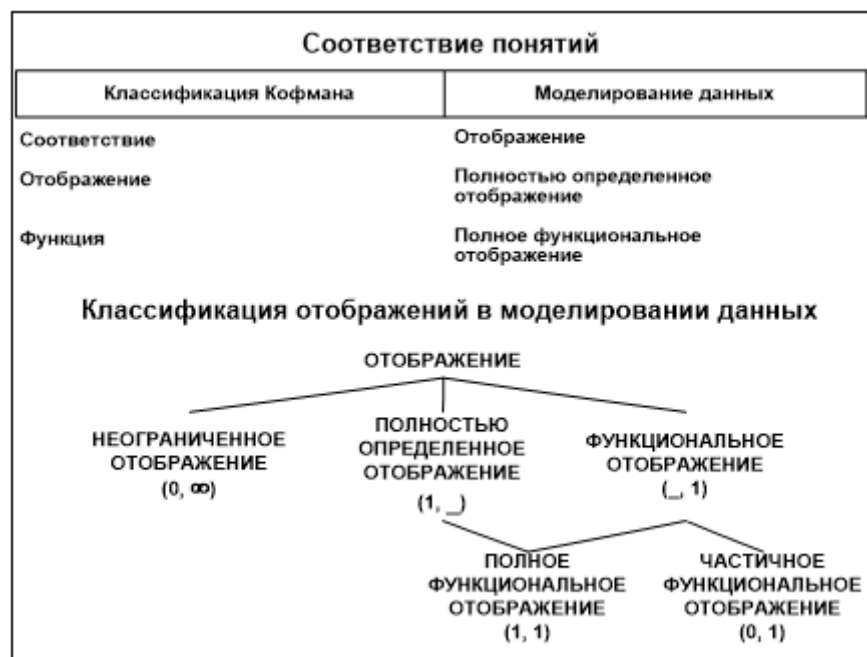
10-11(?)

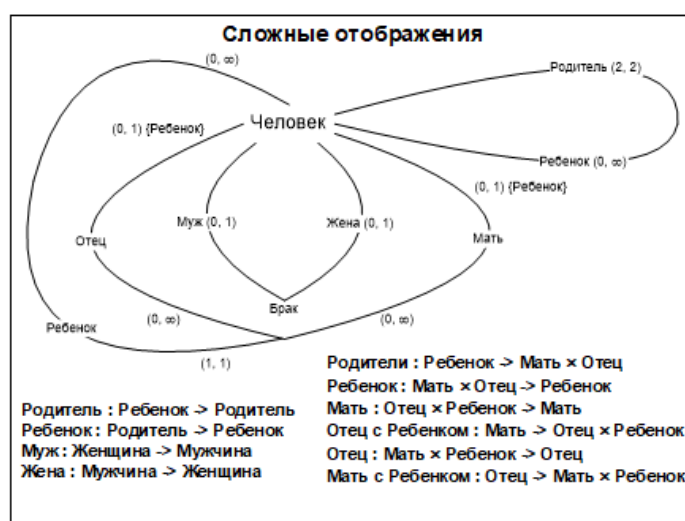
Введенные количественные характеристики отображений позволяют выделить полезные на практике и в теории типы отображений.

Определение 2.3.7. Отображение φ называется:

- неограниченным, если $\hat{I}\hat{e}i\hat{E} \times_{\varphi} = 0$, $\hat{I}\hat{a}\hat{e}\hat{n}\hat{E} \times_{\varphi} = \infty$;
- полностью определенным, если $\hat{I}\hat{e}i\hat{E} \times_{\varphi} \neq 0$;
- функциональным, если $\hat{I}\hat{a}\hat{e}\hat{n}\hat{E} \times_{\varphi} = 1$, в том числе:
 - частичным функциональным, если $\hat{I}\hat{e}i\hat{E} \times_{\varphi} = 0$, $\hat{I}\hat{a}\hat{e}\hat{n}\hat{E} \times_{\varphi} = 1$;
 - полным функциональным, если $\hat{I}\hat{e}i\hat{E} \times_{\varphi} = 1$, $\hat{I}\hat{a}\hat{e}\hat{n}\hat{E} \times_{\varphi} = 1$.

Определение 2.3.8. Говорят, что бинарное отношение, определяющее пару функциональных отображений, есть отношение «**один к одному**» (**1:1**), бинарное отношение с одним функциональным, а другим нефункциональным отображениями есть отношение «**один ко многим**» (**1:M**) или «**многие к одному**» (**M:1**) и бинарное отношение, определяющее пару нефункциональных отображений, есть отношение «**многие ко многим**» (**M:N**).





Определение 2.3.14. До сих пор мы ограничивались рассмотрением **простых отображений**, у каждого экземпляра которых есть один объект-прообраз и один объект-образ.

Однако иногда возникает необходимость в использовании **сложных отображений**, каждый экземпляр которых имеет более одного объекта-прообраза и/или объекта-образа.

Классическим примером такой ситуации в моделировании данных является «поставка некоторым поставщиком детали для конкретного проекта». Каждый факт такой поставки представляет собой тернарную связь между конкретными поставщиком, деталью и проектом.

13

Определение 2.3.15. Если атрибут (группа атрибутов) отношения функционально определяет все другие атрибуты этого отношения, этот атрибут (группа атрибутов) называется **возможным ключом (потенциальным ключом, ключом-кандидатом)** отношения. Слова «функционально определяет» означают, что отображения между возможным ключом и любым атрибутом отношения функциональны.

Определение 2.3.16. Если атрибут (группа атрибутов) отношения функционально определяет отношение, этот атрибут (группа атрибутов) называется **возможным ключом (потенциальным ключом, ключом-кандидатом)** отношения. Т.е. отображение между возможным ключом и самим отношением функционально. Другими словами, возможный ключ уникально идентифицирует кортежи отношения.

Определение 2.3.17. Если атрибут (группа атрибутов) отношения не имеет значений-дубликатов в кортежах отношения, этот атрибут (группа атрибутов) называется **возможным ключом (потенциальным ключом, ключом-кандидатом)** отношения.

Опасность применения этого определения заключается в том, что в отличие от двух предыдущих определений мы анализируем не общий закон Про, а конкретный прецедент, представленный экстенсией отношения. Действительно, из того, что в настоящий момент среди значений атрибута *Фамилия* нет дубликатов, вовсе не следует, что этот атрибут — ключевой. Надо осознавать, что в будущем могут появиться служащие-однофамильцы, нарушающие это определение. Об этом надо всегда помнить и лучше использовать первые два определения.

14

Определение 2.3.18. Если между первичным ключом отношения и некоторым другим атрибутом этого же отношения отображение полностью определено (а точнее, оно – полное функциональное), этот атрибут **не может иметь неопределенных значений** в кортежах отношения. А поскольку между первичным ключом и отношением существует взаимнооднозначное соответствие (каждому значению первичного ключа соответствует один и только один кортеж и наоборот), можно дать следующее определение-следствие.

Определение 2.3.19. Если между отношением и некоторым его атрибутом отображение полностью определено (а точнее, оно – полное функциональное), этот атрибут **не может иметь неопределенных значений** в кортежах отношения.

15(?)

В заключение рассмотрения множеств G_s (правил порождения структур данных), G_c (правил порождения ограничений целостности) и в целом множества G (правил порождения схем), отметим, что понятия интенционала и экстенционала применимы и ко всей БД.

Определение 2.3.20. **Интенционал БД** представляет собой схему БД, включающую определения структур данных и ограничения целостности. **Экстенционал БД** составляют реализации определенных в схеме БД форм данных – множеств и

16 отношений.

ГЛАВА 2.4

Вопросы и задания к параграфу 2.4

1. Дайте определение «расширенного» состояния БД.
2. Что такое операция над данными?
3. Какие два компонента можно выделить в любой операции над данными?
4. Какие действия над данными предусмотрены в технологии БД?
5. Какими способами можно селектировать данные для выполнения тех или иных действий с ними?
6. Какие два класса операций и языков манипулирования данными выделяются в технологии БД? Каковы их особенности?
7. Что такое процедура БД?
8. Какие виды процедур БД вам известны?

1(?)

Динамические аспекты данных отражаются изменением состояния БД. Рассмотрим, например, последовательную выборку строк таблицы с использованием команды «дать следующую строку». Текущее состояние БД определяется совокупностью значений хранящихся данных, а также значением индикатора текущей, указывающим на строку, к которой осуществлялось последнее обращение. Выполнение операции «дать следующую строку» не приведет к изменению реализации БД, но состояние БД изменится, так как изменится значение индикатора текущей.

Определение 2.4.1. **Операции над данными**, выражаемые средствами языка манипулирования данными, переводят БД из состояния DBS_i в состояние DBS_{i+1} (или в неопределенное состояние), причем изменение состояния происходит или при изменении реализации данных, или при изменении управляющих элементов.

Каждое состояние БД должно соответствовать ее схеме. Это означает, что выполнение операций не должно приводить к нарушению свойств данных, отраженных в схеме.

Одним из важных моментов определения операций в моделировании данных является обеспечение достаточной простоты их усвоения пользователем.

2

Определение 2.4.2. Операции обычно задаются в терминах селекции и действия. **Действие** определяет характер операции, а **селекция** – критерий отбора данных, над которыми должно быть произведено действие.

Вполне естественно потребовать, чтобы объектом операции была ограниченная часть БД. Это требование, с одной стороны, исходит из соображений удобства

68

пользователей, а с другой – из целесообразности ограничения числа обращений к внешней памяти, выполняемых при обработке каждого конкретного запроса к БД. Указанное ограничение достигается селекцией данных. Селекция должна осуществляться вне зависимости от того, какая конкретная операция реализуется. Обычно вопросы селекции данных ассоциируются с доступом к данным. Причина этого кроется в архитектуре ЭВМ, в соответствии с которой выполнению операций должен предшествовать перенос данных в оперативную память.

Определение 2.4.3. В технологии БД выделяют пять основных видов действий над данными:

- а) установка текущих – типичные ключевые слова – *SET CURRENCY, FIND*;
- б) включение (добавление новых элементов данных в БД) – *INSERT, ADD*;
- в) обновление (модификация существующих элементов данных в БД) – *UPDATE, MODIFY*;
- г) удаление (исключение элементов данных из БД) – *DELETE, REMOVE*;
- д) выборка (получение данных из БД) – *SELECT, GET*.

Определение 2.4.4. Селекция может осуществляться посредством:

- а) логической позиции в БД – **селекция по текущей**;
- б) значений данных – **селекция по данным**;
- в) связей между данными – **селекция по связям**.

Данные можно селектировать, базируясь на их логических позициях в множествах, отношениях и т.п. Хотя модель может и не поддерживать понятие упорядоченности данных, компьютерная реализация в конечном итоге определяет некоторый порядок. Такой порядок может быть использован для селекции по логической позиции, например, для селекции первого (*FIRST*), последнего (*LAST*), следующего (*NEXT*), предыдущего (*PREVIOUS*) или *n*-го элемента. Назовем этот тип селекции селекцией по текущей.

Определение 2.4.5. Существенный признак, по которому различаются операции и языки данных, определяется характером результата единичной директивы селекции, осуществляемой в ходе выполнения одной операции. Если результат всегда представлен единственным элементом (в зависимости от модели это могут быть значение атрибута, кортеж, сущность, связь и т.д.), полученным при прохождении по логическому пути (т.е. при навигации) в структуре БД, то соответствующие операции называются **навигационными**. Напротив, операции могут специфицировать как бы новую подсхему, определяемую на схеме БД, которой в общем случае соответствует множество элементов, существующих в БД. Такие операции называются **спецификационными**.

Язык, все операции которого являются навигационными, называется **навигационным языком**. Язык, все операции которого являются **спецификационными**, называется **спецификационным языком**. Однако встречаются языки, которые могут иметь как те, так и другие операции.

На слайде представлены реализации запроса «*Выдать фамилии студентов факультета информатики*» на навигационном и **спецификационном** языках. В верхней части слайда приведена схема БД, в рамках которой реализуется запрос. В **графовой** форме представлены три типа объектов (**ФАКУЛЬТЕТ**, **ГРУППА**, **СТУДЕНТ**) и два типа бинарных связей с очевидной семантикой (поэтому их имена мы опустили). Пометки на ребрах говорят о типах этих связей (обе они имеют тип «один ко многим»).

Ниже представлены схемы соответствующих отношений. Как видим, каждому типу объектов соответствует свое отношение, а связи между кортежами-объектами представлены дублированием значений первичных ключей. Понятно, что отношение **ФАКУЛЬТЕТ** понадобится нам, чтобы найти в нем кортеж, относящийся к факультету информатики. Из отношения **СТУДЕНТ** мы получим искомые значения атрибута *Фамилия*. Поскольку напрямую кортежи этих отношений не связаны, нам понадобится отношение **ГРУППА**, имеющее связи с обоими другими отношениями.

Для реализации навигационного запроса нам необходимо по одному разу просканировать кортежи отношений **ФАКУЛЬТЕТ** и **ГРУППА** и перебрать кортежи отношения **СТУДЕНТ** столько раз, сколько групп имеется на факультете информатики. Каждая операция **селектирует** по одному кортежу, делая его текущим установкой на него соответствующего указателя. Пусть вас не вводит в заблуждение отсутствие типичных для навигационных операций ключевых слов *FIRST*, *LAST*, *NEXT*, *PREVIOUS* у первой операции. Очевидно, что атрибут *Название* отношения **ФАКУЛЬТЕТ** является ключевым, и поэтому кортеж с указанным значением этого атрибута один единственный.

Определение 2.4.6. Процедура базы данных представляет собой последовательность операций, выполняемых при определенных условиях. Общность этих условий обеспечивает возможность автоматического, без вмешательства пользователя, инициирования процедур. Действия, выполняемые процедурой, могут быть весьма разнообразны и не ограничиваться частью БД. Процедура посредством информации состояния «уведомляет» систему и пользователя об успешном или неуспешном завершении.

Процедуры БД специфицируются в схеме и в общем случае состоят из трех разделов: условия, действия и уведомления. В разделе условия специфицируется проверка достоверности предпосылок действия. Назначение раздела действий следует из названия. В разделе уведомления указывается, что именно должна передавать процедура при возврате управления и/или задается условие возврата.

Один из видов процедур БД – вычисление значений, которые непосредственно не хранятся в БД, например, вычисление сумм, подсчет числа экземпляров, определение минимума и максимума. Мы будем называть соответствующие процедуры БД **функциями агрегирования**. Во многих моделях функции агрегирования являются конструкциями языка данных.

Второй вид процедур БД – вычисление значения атрибута, например, вычисление возраста человека по заданной текущей дате и дате рождения, что выполняется по запросу о возрасте. Любая процедура БД, вычисляющая значение атрибута, называется **виртуальным атрибутом**. С точки зрения использования виртуальный атрибут обладает

теми же свойствами, что и любой другой атрибут. Однако с ним могут быть связаны различные побочные эффекты и особые ситуации.

Третий вид процедур БД – контроль целостности БД. Верификация ограничений, рассмотренных в предыдущем параграфе, может осуществляться автоматически (декларативный способ определения ОЦ) или с помощью процедур БД, вызываемых явной директивой пользователя или запускаемых при возникновении специфических ситуаций или выполнении специфических операций. Процедуры БД для проверки и/или поддержки ОЦ носят название **триггеров целостности**. Такие процедуры не вырабатывают значений данных, а только «уведомляют» систему об успешном или неуспешном завершении или вызывают принудительное возвращение БД в целостное состояние, согласующееся с определенными ограничениями (что можно трактовать как некоторый побочный эффект).

Четвертый вид процедур БД – обеспечение контроля над доступом. Контроль над доступом не сводится только к парольной защите. Он может быть связан с анализом данных и проверкой полномочий и аутентичности пользователя. Соответствующая процедура БД может выполнять сложные вычисления, по результатам которых могут приниматься важные решения. Например, процедура БД может получать в качестве исходных данных идентификационные параметры пользователя, время, дату, вид действия и вырабатывать сообщения для пользователя, журнала или лица, ответственного за соблюдение правил доступа. Процедуры такого рода называются **триггерами безопасности**.

Процедуры БД могут использоваться также для расширения языка данных операциями, первоначально в нем не предусмотренными. Например, операция сортировки может быть определена как процедура БД, предшествующая выдаче результата спецификационной операции. Другой пример – процедура удаления кортежей-дубликатов, позволяющая модифицировать спецификационную операцию, в результате которой образуется отношение. Такие процедуры называются **операторами доступа**.

Можно определить процедуры БД, которые инициируются операциями модификации одних данных и выполняют косвенные модификации других данных. Если тип связей между отношениями является полным (хотя бы одно из определяемых им отображений полностью определено), то удаление кортежа может привести к запуску процедуры, удаляющей все кортежи, связанные с удаляемым. Такие процедуры называются **триггерами, запускаемыми включением, удалением или обновлением**.

Процедуры БД могут применяться для сбора статистики или для реализации каких-либо других функций администрирования БД. Такие процедуры носят название **процедур администратора БД**. Они не изменяют данные, но могут модифицировать состояние БД.