

1 ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

1.1 Семантическое моделирование данных

Семантика реальной предметной области должна независимым от модели способом представляться в голове проектировщика. В частности, это относится к упоминавшейся нами проблеме представления ограничений целостности. Процесс проектирования начинается с выделения некоторых существенных для приложения объектов предметной области («сущностей») и выявления связей между этими сущностями. Семантика предметной области представляется в виде инфологической модели, поэтому построение инфологической модели называют семантическим моделированием предметной области.

Потребности проектировщиков баз данных в удобных и мощных средствах моделирования предметной области вызвали к жизни направление семантических моделей данных. Любая развитая семантическая модель данных, как и реляционная модель, включает структурную, манипуляционную и целостную части, главным назначением семантических моделей является обеспечение возможности выражения семантики данных.

Прежде чем мы коротко рассмотрим особенности одной из распространенных семантических моделей, остановимся на их возможных применениях.

Наиболее часто на практике семантическое моделирование используется на первой стадии проектирования базы данных. При этом в терминах семантической модели производится концептуальная схема базы данных, которая затем вручную преобразуется к реляционной (или какой-либо другой) схеме. Этот процесс выполняется под управлением методик, в которых достаточно четко оговорены все этапы такого преобразования.

Менее часто реализуется автоматизированная компиляция концептуальной схемы в реляционную. При этом известны два подхода: на основе явного представления концептуальной схемы как исходной информации для компилятора и построения интегрированных систем проектирования с автоматизированным созданием концептуальной схемы на основе интервью с экспертами предметной области. И в том, и в другом случае в результате производится реляционная схема базы данных в третьей нормальной форме (более точно следовало бы сказать, что автору неизвестны системы, обеспечивающие более высокий уровень нормализации).

Наконец, третья возможность, которая еще не вышла (или только выходит) за пределы исследовательских и экспериментальных проектов, – это работа с базой данных в семантической модели, т.е. СУБД, основанные на семантических моделях данных. При этом снова рассматриваются два варианта: обеспечение пользовательского интерфейса на основе семантической модели данных с автоматическим отображением конструкций в реляционную модель данных (это задача примерно такого же уровня

сложности, как автоматическая компиляция концептуальной схемы базы данных в реляционную схему) и прямая реализация СУБД, основанная на какой-либо семантической модели данных. Наиболее близко ко второму подходу находятся современные *объектно-ориентированные СУБД*, модели данных которых по многим параметрам близки к семантическим моделям.

1.2 Основные понятия инфологической модели

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. *Экземпляр* сущности относится к конкретной вещи в наборе. Например, типом сущности может быть ГОРОД, а экземпляром – Москва, Киев и т.д.

Атрибут – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей (например, ЦВЕТ может быть определен для многих сущностей: СОБАКА, АВТОМОБИЛЬ, ДЫМ и т.д.). Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Примерами атрибутов для сущности АВТОМОБИЛЬ являются ТИП, МАРКА, НОМЕРНОЙ ЗНАК, ЦВЕТ и т.д. Здесь также существует различие между типом и экземпляром. Тип атрибута ЦВЕТ имеет много экземпляров или значений:

Красный, Синий, Банановый, Белая ночь и т.д.,

однако каждому экземпляру сущности присваивается только одно значение атрибута.

Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность. Например, для автомобильного завода цвет – это только атрибут продукта производства, а для лакокрасочной фабрики цвет – тип сущности.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Для сущности Расписание ключом является атрибут Номер_рейса или набор: Пункт_отправления, Время_вылета и Пункт_назначения (при условии, что из пункта в пункт вылетает в каждый момент времени один самолет).

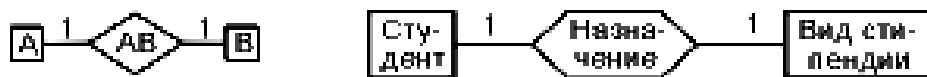
Связь – ассоциирование двух или более сущностей. Если бы назначением базы данных было только хранение отдельных, не связанных между собой данных, то ее структура могла бы быть очень простой. Однако одно из основных требований к организации базы данных – это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи. А так как в реальных базах данных нередко содержатся сотни или даже тысячи сущностей, то теоретически между ними может быть установлено более миллиона связей. Наличие такого множества связей и определяет сложность инфологических моделей.

1.3 Характеристика связей и язык моделирования

При построении инфологических моделей можно использовать язык ER-диаграмм (от англ. Entity-Relationship, т.е. сущность-связь). В них сущности изображаются помеченными прямоугольниками, ассоциации – помеченными ромбами или шестиугольниками, атрибуты – помеченными овалами, а связи между ними – ненаправленными ребрами, над которыми может проставляться степень связи (1 или буква, заменяющая слово «много») и необходимое пояснение.

Между двумя сущностям, например, А и В возможны четыре вида связей.

Первый тип – связь ОДИН-К-ОДНОМУ (1:1): в каждый момент времени каждому представителю (экземпляру) сущности А соответствует 1 или 0 представителей сущности В:



Студент может не «заработать» стипендию, получить обычную или одну из повышенных стипендий.

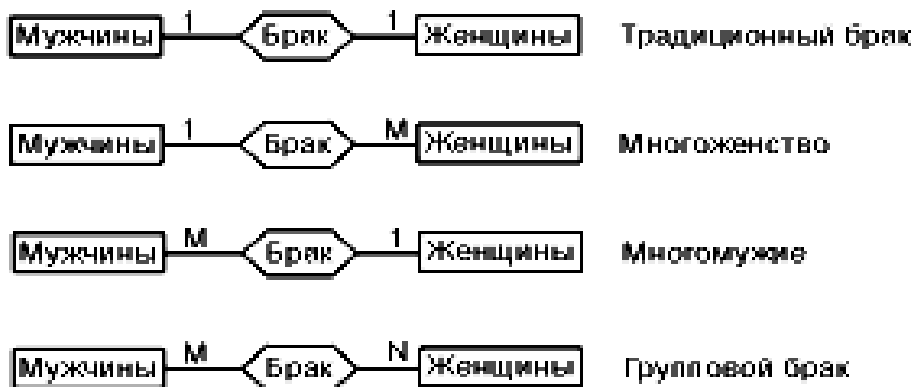
Второй тип – связь ОДИН-КО-МНОГИМ (1:M): одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В.



Квартира может пустовать, в ней может жить один или несколько жильцов.

Так как между двумя сущностями возможны связи в обоих направлениях, то существует еще два типа связи МНОГИЕ-К-ОДНОМУ (М:1) и МНОГИЕ-КО-МНОГИМ (М:N).

Например, если связь между сущностями МУЖЧИНЫ и ЖЕНЩИНЫ называется БРАК, то существует четыре возможных варианта такой связи:



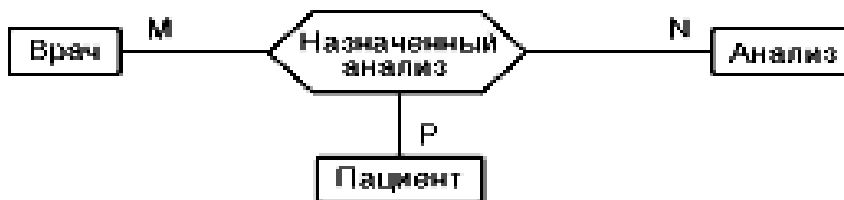
Характер связей между сущностями не ограничивается перечисленными. Существуют и более сложные связи.

Множественные связи между одними и теми же сущностями



Пациент, имея одного лечащего врача, может иметь также несколько врачей-консультантов; врач может быть лечащим врачом нескольких пациентов и может одновременно консультировать несколько других пациентов.

Тренарные связи



Врач может назначить несколько пациентов на несколько анализов, анализ может быть назначен несколькими врачами нескольким пациентам и пациент может быть назначен на несколько анализов несколькими врачами.

Возможны связи более высоких порядков, семантика которых иногда очень сложна.

В приведенных примерах для повышения иллюстративности рассматриваемых связей не показаны атрибуты сущностей и ассоциаций во всех ER-диаграммах. Так, ввод лишь нескольких основных атрибутов в описание брачных связей значительно усложнит ER-диаграмму (рис. 2.1а). В связи с этим язык ER-диаграмм используется для построения небольших моделей и иллюстрации отдельных фрагментов больших.

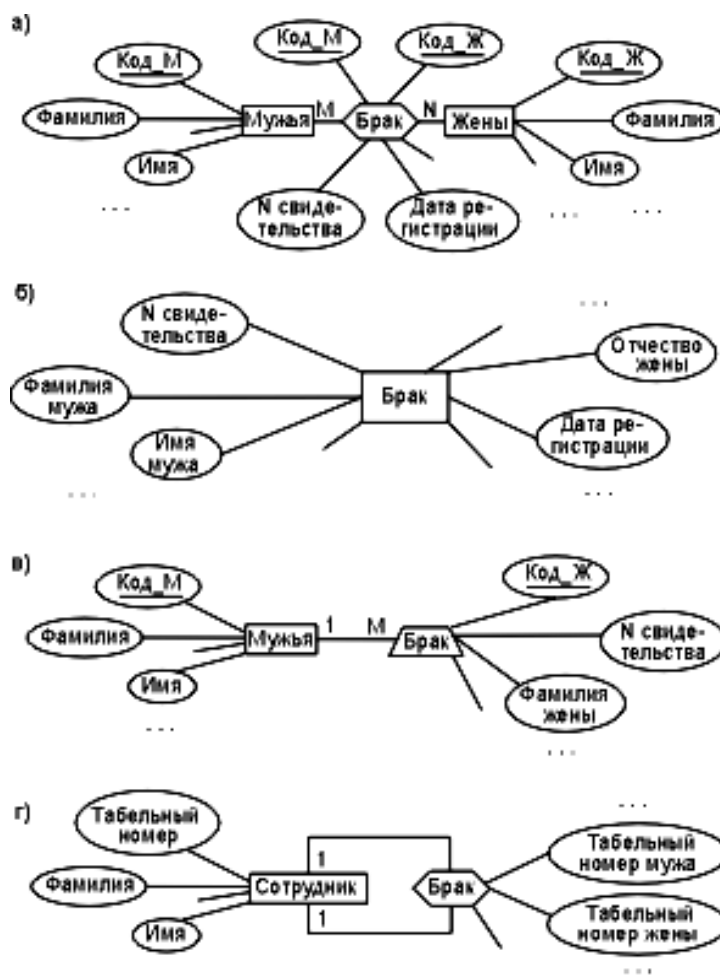


Рис. 2.1. Примеры ER-диаграмм

Чаще же применяется менее наглядный, но более содержательный язык инфологического моделирования (ЯИМ), в котором сущности и ассоциации представляются предложениями вида:

СУЩНОСТЬ (атрибут 1, атрибут 2 , ..., атрибут n)

АССОЦИАЦИЯ [СУЩНОСТЬ S_1 , СУЩНОСТЬ S_2 , ...] (атрибут 1, атрибут 2, ..., атрибут n),

где S – степень связи, а атрибуты, входящие в ключ, должны быть отмечены с помощью подчеркивания.

Так, рассмотренный выше пример множества связей между сущностями может быть описан на ЯИМ следующим образом:

Врач (Номер_врача, Фамилия, Имя, Отчество, Специальность)
 Пациент (Регистрационный_номер, Номер койки, Фамилия,
 Имя, Отчество, Адрес, Дата рождения, Пол)
 Лечащий_врач [Врач 1, Пациент М]
 (Номер_врача, Регистрационный_номер)
 Консультант [Врач М, Пациент N]
 (Номер_врача, Регистрационный_номер).

Для выявления связей между сущностями необходимо, как минимум, определить сами сущности. Но это не простая задача, так как в разных предметных областях один и тот же объект может быть сущностью, атрибутом или ассоциацией. Проиллюстрируем такое утверждение на примерах, связанных с описанием брачных связей.

Отдел записей актов гражданского состояния (ЗАГС) имеет дело не со всеми людьми, а только с теми, кто обратился с просьбой о регистрации брака, рождения или смерти. Поэтому в странах, где допускаются лишь традиционные браки, отделы ЗАГС могут размещать сведения о регистрируемых браках в единственной сущности:

Брак (Номер_свидетельства, Фамилия_мужа, Имя_мужа,
 Отчество_мужа, Дата_рождения_мужа, Фамилия_жены,
 ... , Дата_регистрации, Место_регистрации, ...),

ER-диаграмма которой приведена на рисунке 2.1б.

Теперь рассмотрим ситуацию, когда отдел ЗАГС расположен в стране, допускающей многоженство. Если для регистрации браков использовать сущность «Брак», то будут дублироваться сведения о мужьях, имеющих несколько жен, как показано в таблице 2.1.

Т а б л и ц а 2.1

Номер свидетельства	Фамилия мужа	Фамилия жены	Дата регистрации, г.
1-ЮБ-154745	Петухов ...	Курочкина ...	06/03/1991
1-ЮБ-163489	Петухов ...	Пеструшкина ...	11/08/1991
1-ЮБ-169887	Петухов ...	Рябова ...	12/12/1992
1-ЮБ-169878	Селезнев ...	Уточкина ...	12/12/1992
1-ЮБ-154746	Парасюк ...	Свинюшкина ...	06/03/1991
1-ЮБ-169879	Парасюк ...	Хаврония ...	12/12/1992

Дублирование можно исключить созданием дополнительной сущности «Мужья»

Мужья (Код_М, Фамилия, Имя, Отчество, Дата рождения, Место рождения)

и заменой сущности «Брак» характеристикой со ссылкой на соответствующее описание в сущности «Мужья».

Брак (Номер свидетельства, Код_М, Фамилия жены, ...,
Дата регистрации, ...) {Мужья}.

ER-диаграмма связи этих сущностей показана на рисунке 2.1в, а пример их экземпляров в таблицах 2.2 и 2.3.

Т а б л и ц а 2.2

Код_М	Фамилия Имя Отчество	Год рождения	Место рождения
111	Петухов Альфред Остапович	1971	г. Цапелъка
112	Селезнев Вавила Абрамович	1973	г. Гусев
113	Парасюк Гораций Федулович	1972	г. Свиньин

Т а б л и ц а 2.3

Номер свидетельства	Код_М	Фамилия жены	Имя жены	Дата регистрации
1-ЮБ 154745	111	Курочкина	Августина	06/03/1991
1-ЮБ 163489	111	Пеструшкина	Мариана	11/08/1991
1-ЮБ 169877	111	Рябова	Милана	12/12/1992
1-ЮБ 169878	112	Уточкина	Вероника	12/12/1992 ...
1-ЮБ 154746	113	Свинюшкина	Эльвира	06/03/1991
1 ЮБ 169879	113	Хаврония	Руфина	12/12/1992

Наконец, рассмотрим случай, когда какой-либо организации потребовались данные о наличии в ней семейных пар, а для хранения сведений о сотрудниках уже имеется сущность

Сотрудники (Табельный_номер, Фамилия, Имя, ...).

Использование рассмотренной сущности «Брак» нецелесообразно, т.к. в «Сотрудники» уже есть фамилии, имена и отчества супругов. Поэтому создадим ассоциацию

Брак [Сотрудник 1, Сотрудник 1]
(Табельный_номер_мужа, Табельный_номер_жены, ...),

связывающую между собой определенные экземпляры сущности «Сотрудники» (рис. 2.1г).

Что же такое «связь»? В ER-диаграммах это линия, соединяющая геометрические фигуры, изображающие сущности, атрибуты, ассоциации и другие информационные объекты. В тексте же этот термин используется для указания на взаимозависимость сущностей. Если эта взаимозависимость имеет атрибуты, то она называется ассоциацией.

1.4 Классификация сущностей

Настал момент разобраться в терминологии. К. Дейт определяет три основных класса сущностей: стержневые, ассоциативные и характеристические, а также подкласс ассоциативных сущностей – обозначения.

Стержневая сущность (стержень) – это независимая сущность (несколько подробнее она будет определена ниже).

Ассоциативная сущность (ассоциация) – это связь вида «многие-ко-многим» («-ко-многим» и т.д.) между двумя или более сущностями или экземплярами сущности (как в последнем примере). Ассоциации рассматриваются как полноправные сущности:

- они могут участвовать в других ассоциациях и обозначениях точно так же, как стержневые сущности;
- могут обладать свойствами, т.е. иметь не только набор ключевых атрибутов, необходимых для указания связей, но и любое число других атрибутов, характеризующих связь.

Например, ассоциации «Брак» содержат ключевые атрибуты «Код_М», «Код_Ж» и «Табельный номер мужа», «Табельный номер жены», а также уточняющие атрибуты «Номер свидетельства», «Дата регистрации», «Место регистрации», «Номер записи в книгу ЗАГС» и т.д.

Характеристическая сущность (характеристика) – это связь вида «многие-к-одной» или «одна-к-одной» между двумя сущностями (частный случай ассоциации). Единственная цель характеристики в рамках рассматриваемой предметной области состоит в описании или уточнении некоторой другой сущности. Необходимость в них возникает в связи с тем, что сущности реального мира имеют иногда многозначные свойства. Муж может иметь несколько жен, книга – несколько характеристик переиздания (исправленное, дополненное, переработанное) и т.д.

Существование характеристики полностью зависит от характеризуемой сущности: женщины лишаются статуса жен, если умирает их муж.

Для описания характеристики используется новое предложение ЯИМ, имеющее в общем случае вид:

ХАРАКТЕРИСТИКА (атрибут 1, атрибут 2, ...)
{СПИСОК ХАРАКТЕРИЗУЕМЫХ СУЩНОСТЕЙ}.

Расширим также язык ER-диаграмм, введя для изображения характеристики трапецию (рис. 2.2).

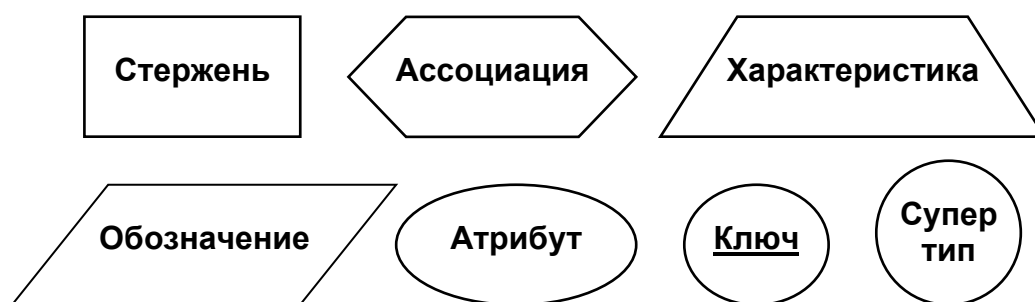


Рис. 2.2. Элементы расширенного языка ER-диаграмм

Обозначающая сущность или обозначение – это связь вида «многие-к-одной» или «одна-к-одной» между двумя сущностями и отличается от характеристики тем, что не зависит от обозначаемой сущности.

Рассмотрим пример, связанный с зачислением сотрудников в различные отделы организации.

При отсутствии жестких правил (сотрудник может одновременно зачисляться в несколько отделов или не зачисляться ни в один отдел) необходимо создать описание с ассоциацией Зачисление:

Отделы (Номер отдела, Название отдела, ...)

Служащие (Табельный номер, Фамилия, ...)

Зачисление [Отделы М, Служащие N] (Номер отдела, Табельный номер, Дата зачисления).

Однако при условии, что каждый из сотрудников должен быть обязательно зачислен в один из отделов, можно создать описание с обозначением Служащие:

Отделы (Номер отдела, Название отдела, ...)

Служащие (Табельный номер, Фамилия, ... , Номер отдела, Дата зачисления)[Отделы]

В данном примере служащие имеют независимое существование (если удаляется отдел, то из этого не следует, что также должны быть удалены служащие такого отдела). Поэтому они не могут быть характеристиками отделов и названы обозначениями.

Обозначения используют для хранения повторяющихся значений больших текстовых атрибутов: «кодификаторы» изучаемых студентами дисциплин, наименований организаций и их отделов, перечней товаров и т.п.

Описание обозначения внешне отличается от описания характеристики только тем, что обозначаемые сущности заключается не в фигурные скобки, а в квадратные:

ОБОЗНАЧЕНИЕ (атрибут 1, атрибут 2, ...)[СПИСОК ОБОЗНАЧАЕМЫХ СУЩНОСТЕЙ].

Как правило, обозначения не рассматриваются как полноправные сущности, хотя это не привело бы к какой-либо ошибке.

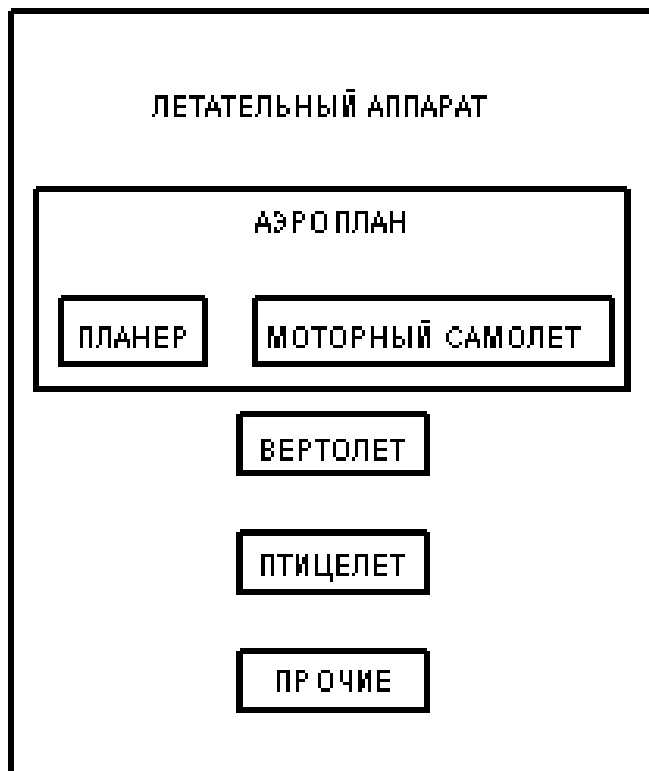
Обозначения и характеристики не являются полностью независимыми сущностями, поскольку они предполагают наличие некоторой другой сущности, которая будет «обозначаться» или «характеризоваться». Однако они все же представляют собой частные случаи сущности и могут, конечно, иметь свойства, могут участвовать в ассоциациях, обозначениях и иметь свои собственные (более низкого уровня) характеристики. Подчеркнем также, что все экземпляры характеристики должны быть обязательно связаны с каким-либо экземпляром характеризуемой сущности. Однако допускается, чтобы некоторые экземпляры характеризуемой сущности не имели связей. Правда, если это касается браков, то сущность «Мужья» должна быть заменена на сущность «Мужчины» (нет мужа без жены).

1.4.1 Супертипы и подтипы

Сущность может быть расщеплена на два или более взаимно исключающих подтипа, каждый из которых включает общие атрибуты и/или связи. Эти общие атрибуты и/или связи явно определяются один раз на более высоком уровне. В подтипах могут определяться собственные атрибуты и/или связи. В принципе подтипизация может продолжаться на более низких уровнях, но опыт показывает, что в большинстве случаев оказывается достаточно двух-трех уровней.

Сущность, на основе которой определяются подтипы, называется *супертипом*. *Подтипы* должны образовывать полное множество, т.е. любой экземпляр супертипа должен относиться к некоторому подтипу. Иногда для полноты приходится определять дополнительный подтип ПРОЧИЕ.

Пример: Супертип ЛЕТАТЕЛЬНЫЙ АППАРАТ



Как полагается это читать? От супертипа: ЛЕТАТЕЛЬНЫЙ АППАРАТ, который должен быть АЭРОПЛАНОМ, ВЕРТОЛЕТОМ, ПТИЦЕЛЕТОМ или ДРУГИМ ЛЕТАТЕЛЬНЫМ АППАРАТОМ. От подтипа: ВЕРТОЛЕТ, который относится к типу ЛЕТАТЕЛЬНОГО АППАРАТА. От подтипа, который является одновременно супертипом: АЭРОПЛАН, который относится к типу ЛЕТАТЕЛЬНОГО АППАРАТА и должен быть ПЛАНЕРОМ или МОТОРНЫМ САМОЛЕТОМ.

Иногда удобно иметь два или более разных разбиения сущности на подтипы. Например, сущность ЧЕЛОВЕК может быть разбита на подтипы по профессиональному признаку (ПРОГРАММИСТ, ДОЯРКА и т.д.), а может – по половому признаку (МУЖЧИНА, ЖЕНЩИНА).

1.5 Первичные и внешние ключи

Напомним, что ключ или возможный ключ – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждая сущность обладает хотя бы одним возможным ключом. Один из них принимается за первичный ключ. При выборе первичного ключа следует отдавать предпочтение несоставным ключам или ключам, составленным из минимального числа атрибутов. Нецелесообразно также использовать ключи с длинными текстовыми значениями (предпочтительнее использовать целочисленные атрибуты). Так, для идентификации студента можно использовать либо уникальный номер зачетной книжки, либо набор из фамилии, имени, отчества, номера группы и, может быть, дополнительных

атрибутов, так как не исключено появление в группе двух студентов с одинаковыми фамилиями, именами и отчествами. Плохо также использовать в качестве ключа не номер блюда, а его название, например «Закуска из плавленных сырков «Дружба» с ветчиной и соленым огурцом» или «Заяц в сметане с картофельными крокетами и салатом из красной капусты».

Не допускается, чтобы первичный ключ стержневой сущности (любой атрибут, участвующий в первичном ключе) принимал неопределенное значение. Иначе возникнет противоречивая ситуация: появится не обладающий индивидуальностью, и не существующий экземпляр стержневой сущности. По тем же причинам необходимо обеспечить уникальность первичного ключа.

1.5.1 Внешние ключи

Если сущность С связывает сущности А и В, то она должна включать внешние ключи, соответствующие первичным ключам сущностей А и В.

Если сущность В обозначает сущность А, то она должна включать внешний ключ, соответствующий первичному ключу сущности А. В п. 2.4 рассматривался пример, где «Служащие» обозначали «Отделы» и включали внешний ключ «Номер отдела», соответствующий первичному ключу сущности «Отделы».

Переопределим теперь стержневую сущность как сущность, которая не имеет внешних ключей, т.е. не является ни ассоциацией, ни обозначением, ни характеристикой.

Связь между первичными и внешними ключами сущностей иллюстрируется на рисунке 2.3.

Здесь для обозначения любой из ассоциируемых сущностей (стержней, характеристик, обозначений или даже ассоциаций) используется новый обобщающий термин «Цель» или «Целевая сущность».

Таким образом, при рассмотрении проблемы выбора способа представления ассоциаций и обозначений в базе данных основной вопрос, на который следует получить ответ: «Каковы внешние ключи?» И далее, для каждого внешнего ключа необходимо решить три вопроса.

1) Может ли данный внешний ключ принимать неопределенные значения (NULL-значения)? Иначе говоря, может ли существовать некоторый экземпляр сущности данного типа, для которого неизвестна целевая сущность, указываемая внешним ключом? В случае поставок это, вероятно, невозможно – поставка, осуществляемая неизвестным поставщиком, или поставка неизвестного продукта не имеют смысла. Но в случае с сотрудниками такая ситуация однако могла бы иметь смысл – вполне возможно, что какой-либо сотрудник в данный момент не зачислен вообще ни в какой отдел. Заметим, что ответ на данный вопрос не зависит от прихоти проектировщика базы данных, а определяется фактическим образом действий, принятым в той части реального мира, которая должна быть представлена в рассматриваемой базе данных. Подобные замечания имеют отношение и к вопросам, обсуждаемым ниже.

2) Что должно случиться при попытке УДАЛЕНИЯ целевой сущности, на которую ссылается внешний ключ? Например, при удалении поставщика, который осуществил, по крайней мере, одну поставку.

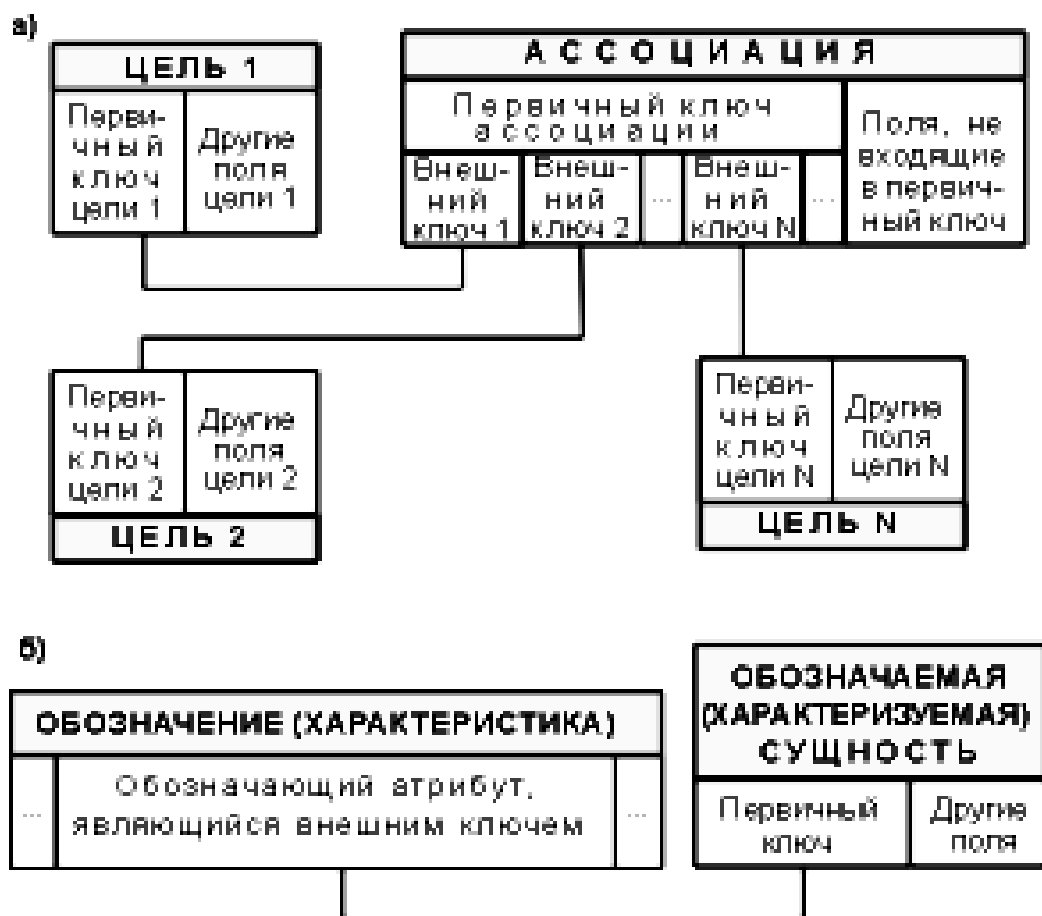


Рис. 2.3. Структуры сущностей:

а – ассоциации; б – обозначения (характеристики)

Существует три возможности.

КАСКАДИРОВАНИЕ – операция удаления «каскадируется» с тем, чтобы удалить также поставки этого поставщика.

ОГРАНИЧЕНИЕ – удаляются лишь те поставщики, которые еще не осуществляли поставок. Иначе операция удаления отвергается.

УСТАНОВКА – Для всех поставок удаляемого поставщика NULL-значение внешний ключ устанавливается в неопределенное значение, а затем этот поставщик удаляется. Такая возможность, конечно, неприменима, если данный внешний ключ не должен содержать NULL-значений.

3) Что должно происходить при попытке ОБНОВЛЕНИЯ первичного ключа целевой сущности, на которую ссылается некоторый внешний ключ? Например, может быть предпринята попытка обновить номер такого поставщика, для которого имеется, по крайней мере, одна соответствующая

поставка. Этот случай для определенности снова рассмотрим подробнее. Имеются те же три возможности, как и при удалении.

КАСКАДИРОВАНИЕ – операция обновления «каскадируется» с тем, чтобы обновить также и внешний ключ в поставках этого поставщика.

ОГРАНИЧЕНИЕ – обновляются первичные ключи лишь тех поставщиков, которые еще не осуществляли поставок. Иначе операция обновления отвергается.

УСТАНОВКА – для всех поставок такого поставщика NULL-значение внешний ключ устанавливается в неопределенное значение, а затем обновляется первичный ключ поставщика. Такая возможность, конечно, неприменима, если данный внешний ключ не должен содержать NULL-значений.

Таким образом, для каждого внешнего ключа проектировщик базы данных должен специфицировать не только поле или комбинацию полей, составляющих этот внешний ключ, и целевую сущность, которая идентифицируется этим ключом, но также и ответы на указанные выше вопросы (три ограничения, которые относятся к этому внешнему ключу).

Наконец, о характеристиках – обозначающих сущностях, существование которых зависит от типа обозначаемых сущностей. Обозначение представляется внешним ключом в таблице, соответствующей этой характеристике. Но три рассмотренных выше ограничения на внешний ключ для данного случая должны специфицироваться следующим образом:

NULL-значения не допустимы

УДАЛЕНИЕ ИЗ (цель) КАСКАДИРУЕТСЯ

ОБНОВЛЕНИЕ (первичный ключ цели) КАСКАДИРУЕТСЯ

Указанные спецификации представляют зависимость по существованию характеристических сущностей.