

3.2. Модель данных «Сущность-Связь» (ER-модель)

Один из наиболее известных и получивших широкое распространение методов семантического моделирования основан на применении модели «Сущность-Связь» (англ. Entity-Relationship, ER), предложенной Питером Пин Шан Ченом (англ. Peter Pin Shan Chen) в 1976 году и с тех пор неоднократно усовершенствованной. ER-модель была задумана как средство представления схемы предметной области, не зависящего от особенностей среды хранения и не связанного соображениями физической эффективности. В модели «Сущность-Связь» используется естественное представление, в соответствии с которым реальный мир состоит из сущностей (предметов) и связей (отношений между предметами).

Коль скоро концепции ER-модели очень близки человеческому восприятию окружающего мира, следовало ожидать реализации этой модели, по крайней мере, в программных средствах автоматизированного проектирования БД (CASE-системах). Так это впоследствии и произошло. Вот только реализации удостоилась не классическая ER-модель Чена, а те ее модификации, которые можно назвать семантическими моделями с большой натяжкой.

Полезной модификацией классической ER-модели Чена является так называемая расширенная ER-модель (англ. Enhanced ER или EER). EER-модель включает все концепции ER-модели Чена плюс дополнительные концепции специализации и категоризации.

Последующие модификации ER-модели (нотации Баркера и IDEF1X), хоть и удостоились реализации в CASE-инструментах (Oracle Designer и ERwin соответственно), шли по пути сокращения выразительных возможностей представления семантики предметной области. Последняя модификация ER-модели (IDEF1X) отличается от реляционной модели, по сути, взаимнооднозначным переименованием понятий: «тип сущностей» – «отношение», «тип связей» – «ограничение ссылочной целостности», «атрибут типа сущностей» – «атрибут отношения».

В первых трех подразделах этого параграфа мы последовательно познакомимся со структурными понятиями, ограничениями целостности и операциями основополагающей ER-модели Чена. В четвертом подразделе мы обсудим основное назначение семантических моделей данных на примере ER-модели. В последнем подразделе мы охарактеризуем другие широко распространенные варианты ER-модели.

3.2.1. Структуры

Структурные понятия ER-модели Чена	
Множество сущностей (МСу) – E	<i>СЛУЖАЩИЙ</i>
Множество связей (МСв) – R	
$R \subset \{ \langle e_1, e_2, \dots, e_n \rangle \mid e_1 \in E_1 \wedge e_2 \in E_2 \wedge \dots \wedge e_n \in E_n \}$	
$УПРАВЛЕНИЕ = \{ \langle e_1, e_2 \rangle \mid e_1 \in СЛУЖАЩИЙ \wedge e_2 \in СЛУЖАЩИЙ \}$	
Роль - r	
$R \subset \{ \{ r_1 : e_1, r_2 : e_2, \dots, r_n : e_n \} \mid e_1 \in E_1 \wedge e_2 \in E_2 \wedge \dots \wedge e_n \in E_n \}$	
$УПРАВЛЕНИЕ = \{ \{ Начальник : e_1, Подчиненный : e_2 \} \mid e_1 \in СЛУЖАЩИЙ \wedge e_2 \in СЛУЖАЩИЙ \}$	
Множество значений (МЗн) – V	<i>ЦПЧ или Строки</i>
Атрибут - f	
$f : E \rightarrow V$	<i>Фамилия: ПАЦИЕНТ -> Строки</i>
$f : R \rightarrow V$	<i>Номер койки: РАЗМЕЩЕНИЕ -> ЦПЧ</i>
$f : E \rightarrow V_1 \times V_2 \times \dots \times V_n$	<i>Адрес: ПАЦИЕНТ -> Строки x ЦПЧ</i>
$f : R \rightarrow V_1 \times V_2 \times \dots \times V_n$	<i>Серия_Номер_Свидетельства: БРАК -> ЦПЧ x ЦПЧ</i>

Определение 3.2.1. Сущность (англ. entity) – это абстрактное представление единичного объекта ПрО. Эта абстракция строится на основании некоторого понятия о предметах, существенного с точки зрения задач ПрО. Любое понятие характеризуется содержанием (условием, истинность которого говорит о том, что объект подпадает под это понятие) и объемом (классом объектов, удовлетворяющих условию содержания). Содержание понятия предполагает наличие определенных характеристик, общих для всех объектов объема, возможно различающихся значениями. Очевидно, что с точки зрения различных понятий можно построить несколько абстракций-сущностей одного и того же объекта. Каждое понятие определяет свое **множество сущностей** (англ. entity set), интенционалом которого является содержание понятия, а экстенционалом – его объем.

Определение 3.2.2. Связь (англ. relationship) – это единичный экземпляр отношения (в философско-логическом смысле) между сущностями. Абсолютно все, что говорилось о сущности и множестве сущностей, можно сказать о связи и множестве связей. Разница заключается лишь в том, что для последних используется не понятие о предметах (как у сущностей), а понятие об *n*-ках предметов. **Множество связей** (англ. relationship set) можно рассматривать как математическое отношение, определенное в общем случае на *n* множествах сущностей, а связь – как кортеж этого отношения с *n* элементами-сущностями.

Определение 3.2.3. Каждая сущность в связи играет определенную **роль** (англ. role) – функцию, общую для всех сущностей этой роли. Таким образом, даже если в понятии об *n*-ках предметов неоднократно участвует одно и то же множество сущностей, в каждой связи сущности этого типа будут различаться своими ролями. В силу того, что множество связей рассматривается как математическое отношение, порядок упоминания сущностей в связи является значимым, и поэтому в первом случае, показанном на слайде, для представления связей используются кортежи в смысле математики. Однако указание ролей сущностей в связях делает соблюдение порядка сущностей необязательным, и поэтому во втором случае для представления связей используются уже множества пар «*роль : сущность*» (кортежи в смысле моделирования данных).

Определение 3.2.4. Информацию об объекте или взаимоотношении между объектами получают путем наблюдения или измерения характеристик, существенных для того или иного понятия, и выражают множеством пар «*атрибут – значение*». **Значения** (англ. value) классифицируются в различные **множества значений** (англ. value set). Таким образом, множество значений в этой модели представляет собой то, что в других моделях иногда называют доменом, – область допустимых значений атрибутов.

Определение 3.2.5. Атрибут (англ. attribute) в ER-модели формально определяется как отображение, ставящее в соответствие сущностям или связям некоторого множества одиночные значения или кортежи значений. Таким образом, областью определения атрибутного отображения может быть множество сущностей или множество связей, а областью значений – множество значений или Декартово произведение множеств значений. Причем это отображение не обязано быть функциональным, образов у одного прообраза может быть несколько. В таком случае говорят о **многозначных атрибутах**. Обратите внимание, что в этой модели в форме атрибутов представляются как свойства, так и характеристики сущностей и связей. У первых просто фиксирована ОЗО – множество истинностных значений (*Истина* и *Ложь*), в то время как у вторых она произвольна.

Мы познакомились со структурными понятиями ER-модели. Заметим только, что множества сущностей, множества связей, роли, множества значений и атрибуты являются типами и составляют схему БД в этой модели, а сущности, связи и значения – суть, знаки, и представляют данные. Отметим, что в ER-модели для связывания сущностей используются не значения атрибутов (как в реляционной модели), а специальные уникальные указатели. Кстати, это присуще всем моделям данных, кроме реляционной.

Как видим, предлагаемые в ER-модели структурные понятия очень близки по смыслу общечеловеческим понятиям: сущность – предмет, связь – отношение (в философско-логическом смысле), атрибут – свойство или характеристика. Надо только отдавать себе отчет, что происходит переход от интуитивно воспринимаемых понятий к формальным понятиям со своими определениями и свойствами, которые частично обсуждались во второй главе.



ER-модель предоставляет замечательную графическую форму представления схемы БД – **ER-диаграмму** (англ. Entity-Relationship Diagram, ERD). С точки зрения наглядности отражения семантики данных она не имеет себе равных (один рисунок порой стоит тысячи слов). Для своего варианта ER-модели Чен предложил следующую графическую нотацию.

Схема БД – граф, вершинами которого являются множества сущностей, множества связей и множества значений. Эти вершины представлены различными графическими примитивами: множества сущностей – прямоугольниками, множества связей – ромбами, множества значений – овалами. Имена множеств указываются внутри графических фигур.

Неориентированные ребра соответствуют ролям множеств сущностей во множестве связей и соединяют эти множества. Ориентированные дуги представляют атрибутные отображения. Они выходят из вершины множества сущностей или множества связей и входят в одно или несколько (в этом случае дуга разделяется на несколько концов) множеств значений. Ребра по необходимости помечаются именами ролей, а дуги всегда помечаются именами атрибутов.

Несмотря на то, что в нотации предусмотрены выразительные возможности для всех структурных элементов схемы, часто множества значений и атрибуты на ER-диаграмме не показывают, а выносят их в традиционный текстовый синтаксис, подобный использовавшемуся в параграфе 2.5.

Пример ER-диаграммы для нашей демонстрационной медицинской ПрО, в которой представлен пока только структурный аспект, приведен на слайде (множества значений и атрибуты опущены). На ней в компактной и наглядной форме показаны все типы объектов ПрО и их взаимосвязи. Если к тому же удачно подобраны имена всех структурных компонентов схемы, никаких дополнительных комментариев часто не требуется.

В ER-диаграмме хорошо проявляется информативность схемы – способность БД, по ней построенной, предоставлять ту или иную информацию. В нашей схеме, в частности, хорошо просматриваются три пути между множествами сущностей *ПАЦИЕНТ* и *БОЛЬНИЦА*. Пройдя от конкретного пациента по связям левого маршрута (через множество сущностей *ПАЛАТА*), мы можем получить сущность стационара, в котором он лежит. Средний путь (через множество сущностей *ВРАЧ*) приведет нас в больницы, в штате которых числятся врачи, имеющие связи с нашим пациентом. Наконец, правый маршрут (через множества сущностей *АНАЛИЗ* и *ЛАБОРАТОРИЯ*) даст нам множество больниц, которые могут обслуживаться лабораториями, обрабатывавшими анализы нашего пациента.

В дальнейшем будут приведены фрагменты ER-диаграмм, включающие множества значений, атрибуты множеств сущностей и множеств связей, а также множества связей степени больше двух.

Но, прежде всего надо обсудить уже упоминавшуюся в параграфе 3.1 проблему триализма. Первый раз мы рассмотрим ее на примере ER-модели.



Выбирая форму данных, помните о возможном **триализме** (наличии трех форм представления) явлений моделируемого мира. Так, брак между людьми можно мыслить как атрибут, множество связей или множество сущностей. Проектируя схему БД, необходимо сделать выбор в пользу одного из этих представлений. При этом можно использовать следующие соображения.

Если вам достаточно информации о том, состоит человек в браке или нет, можно рассмотреть это явление как атрибут множества сущностей *ЧЕЛОВЕК*. Если вас к тому же интересует, с кем конкретно заключен брак, необходимо трактовать *БРАК* как множество связей между сущностями множества *ЧЕЛОВЕК*. Даже если вы захотите описать это явление с помощью тех или иных характеристик, вам не придется менять форму множества связей (в ER-модели Чена множества связей могут иметь характеристики). А вот если вам понадобится представлять связи браков с другими явлениями, вам не обойтись без множества сущностей *БРАК*.

Как вы понимаете, мы перечислили варианты представления в порядке возрастания функциональности и информативности форм. Но это не означает, что все явления нужно трактовать как сущности. Бинаризация представлений (сведение структур данных к понятиям бинарной модели – множествам знаков и исключительно бинарным отношениям между ними), на первый взгляд, упрощает проектирование, но она скрадывает важную для проектирования эффективной схемы БД семантику ПрО, невозполнимую на последующих этапах. Следует отметить, что в данном случае проблема бинаризации выглядит не очень показательной. Подробнее на анализе этой проблемы мы остановимся в параграфе 3.3.

Первые два варианта реализации явлений типа «брак» представлены в левой части слайда (вверху – текстовые определения, внизу – ER-диаграммы) и, по-видимому, в пояснениях не нуждаются. В третьем случае (правая часть слайда) предполагается, что помимо сведений о состоянии людей в браке интерес представляет информация о том, кто рожден в момент зарегистрированного брака между родителями. Налицо необходимость в констатации наличия связей типа *РОЖДЕНИЕ В БРАКЕ* между множествами сущностей *ЧЕЛОВЕК* и *БРАК*.

Обратите внимание на то, что в связи могут вступать только сущности (а не связи или значения), и поэтому явления типа «брак» в последнем случае мы вынуждены представлять в виде множества сущностей. В таком случае роли *Муж* и *Жена* (второй случай) трансформируются в множества связей *МУЖ В БРАКЕ* и *ЖЕНА В БРАКЕ*.

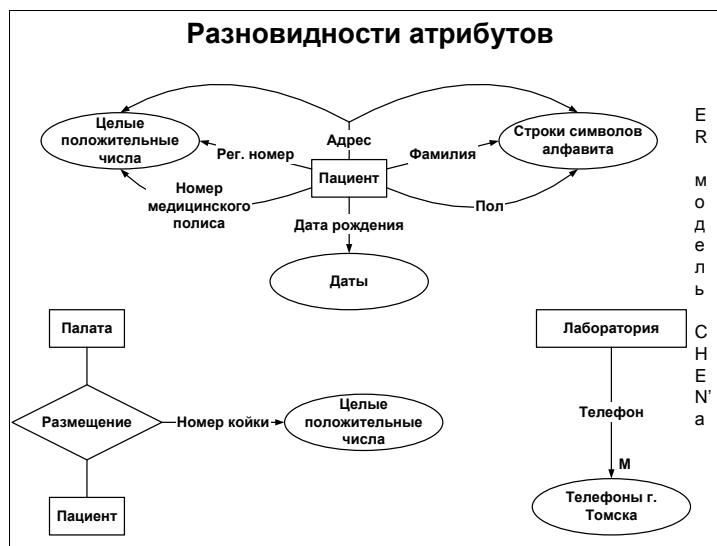


На слайде приведены дополнительные примеры разновидностей множеств связей, не нашедших отражения на ER-диаграмме медицинской ПрО.

Необязательно, чтобы все множества сущностей, на которых определено множество связей, были различными. Например, рекурсивное множество связей *УПРАВЛЕНИЕ* может быть определено на множестве сущностей *СЛУЖАЩИЙ*. Другим примером рекурсивного бинарного множества связей является множество связей *РОЖДЕНИЕ*, определяющее взаимоотношения типа «родитель-ребенок» между парой людей. В верхней правой части слайда показано рекурсивное тернарное множество связей *РОЖДЕНИЕ*, каждая связь которого определяет взаимоотношения между тройкой людей – отцом, матерью и их ребенком. Кстати, именно в случае рекурсивных множеств связей особенно актуальными становятся пометки в виде имен ролей на ребрах, соединяющих множества сущностей с множеством связей.

Между одними и теми же множествами сущностей иногда бывает более одного множества связей. Возможности адекватного представления этой ситуации, которые обеспечиваются ER-моделью, показаны в нижней части слайда. В отличие от приведенной ранее ER-диаграммы нашей демонстрационной ПрО здесь определены специализированные множества связей между врачами и пациентами – *НАПРАВЛЕНИЕ НА АНАЛИЗ*, *КОНСУЛЬТАЦИЯ*, *ЛЕЧАЩИЙ ВРАЧ* и *ПОСТАНОВКА ДИАГНОЗА*. Как видим два из них бинарны, а два – тернарны.

На сравнении этих двух вариантов представления связей между врачами и пациентами обсудим проблему информативности схемы данных. Надеюсь, всем очевидно, что второй вариант (со специализированными связями) обеспечивает пользователей более детальной информацией об этих связях. Менее очевидно преимущество этого варианта в представлении информации об анализах пациентов. В условиях основной ER-схемы медицинской ПрО (приведенной ранее) мы можем получить информацию о том, что пациент делал анализ, и о том, что он имел какие-то контакты с врачами (возможно, не только по поводу анализов). Ответа на вопрос, «Какой врач послал пациента на конкретный анализ?» мы не получим. Новая схема, благодаря наличию тернарного множества связей *НАПРАВЛЕНИЕ НА АНАЛИЗ*, обеспечит нас и такой информацией. Обычно чуда не происходит, и большая информативность схемы сопряжена с дополнительными расходами памяти. Для данной конкретной ситуации, как ни странно, это не характерно.



Приведенный слайд демонстрирует представление в ER-диаграмме атрибутивных отображений множества сущностей *ПАЦИЕНТ* и множества связей *РАЗМЕЩЕНИЕ*.

Атрибут обеспечивает интерпретацию множества значений (а значит, и самих значений) в контексте множества сущностей или множества связей. Например, атрибут *Дата рождения* уточняет семантику использования множества значений *ДАТЫ* в контексте множества сущностей *ПАЦИЕНТ*.

Примером атрибутивного отображения множества сущностей в Декартово произведение множеств значений является атрибут *Адрес*, у которого областью значений является произведение множеств значений *СТРОКИ СИМВОЛОВ АЛФАВИТА* и *ЦЕЛЫЕ ПОЛОЖИТЕЛЬНЫЕ ЧИСЛА*. Первое множество обеспечивает значения названий улиц, второе – номеров домов.

Некоторые атрибуты могут быть многозначными. Например, атрибут *Телефон* множества сущностей *ЛАБОРАТОРИЯ* может иметь более чем одно значение для каждой лаборатории. В этом случае в ER-диаграмме конец атрибутивной дуги помечается символом «М». Такая же пометка сопровождает многозначные атрибуты и в текстовой форме определения атрибутов. Аналогичным свойством обладает также атрибут *Телефон* множества сущностей *БОЛЬНИЦА*. Впрочем, эти свойства атрибутов относятся скорее к ограничениям целостности.

Атрибуты могут быть не только у множеств сущностей, но и множеств связей. Так, например, атрибут *Номер койки* может быть непосредственно ассоциирован с множеством связей *РАЗМЕЩЕНИЕ*.

Обратите внимание, насколько естественно такое представление этой характеристики. Очевидно, что она относится явно не к палатам (для каждой из них характерно множество таких номеров) и не к пациентам (большинству людей она вовсе не характерна). И именно в тот момент, когда пациент размещается в палате, и создается новая связь типа *РАЗМЕЩЕНИЕ*, она становится актуальной для этой связи. И это продолжается до тех пор, пока пациент не уйдет из этой палаты (выпишется из больницы или перейдет в другую палату), и связь не будет удалена.



Автор ER-модели Чен предусмотрел для своей модели две формы представления экстенционала БД – графовую и табличную.

Вершинами графа знаков в ER-модели, как и следовало ожидать, являются сущности, связи и значения. Для удобства их интерпретации принято, во-первых, объединять элементы одного множества (множества обведены пунктирной линией), во-вторых, пометать множества их именем и, в-третьих, располагать однотипные множества в соответствующих мысленно организуемых столбцах (на слайде они обозначены словами *Множества сущностей*, *Множества связей*, *Множества значений*). Вершины сущностей и связей помечены их условными знаками ($e1$, $e2$, r), вершины значений – самими значениями.

Неориентированные ребра между сущностями и связями, соответствуют ролям сущностей в связи. Имена ролей, помечающие эти ребра, располагаются в столбце с именем *Роли*. Наконец, ориентированные дуги, выходящие из вершин-сущностей или вершин-связей и входящие в вершины-значения, представляют экземпляры атрибутивных отображений. Пометка на дуге в столбце *Атрибуты* задает имя атрибута.

Отметим, что, как и ранее, в графе экстенционала БД вершины-значения не дублируются для каждого экземпляра атрибутивного отображения, образом которого является это значение.

Экстенционал БД в форме таблиц				
Множеств сущностей	Атрибуты			
	Сущности	Номер палаты	Наименование	Число коек
	ПАЛАТА	Целые числа	Строки символов	Целые числа
	e1	111	Резанимационная	1
	e5	555	После-операционная	2
Множеств связей
	Содержать пациента		Лежать в палате	
	РАЗМЕЩЕНИЕ			Номер койки
	ПАЛАТА	ПАЦИЕНТ	Целые числа	
	e1	e2	1	
	e5	e10	1	
	

Экстенционал БД в ER-модели может быть представлен и в табличной форме. Каждому множеству сущностей и множеству связей соответствует своя таблица. На слайде представлена таблица для множества сущностей *ПАЛАТА* и таблица для множества связей *РАЗМЕЩЕНИЕ*.

Как видим, в таблице множества сущностей предусмотрена двухстрочная шапка, обеспечивающая полную интерпретацию сущностей и значений их атрибутов. Верхняя строка шапки задает имена атрибутов, значения которых будут располагаться в столбцах таблицы, начиная со второго. Вторая строка шапки именует множество сущностей (первый столбец) и определяет множества значений для соответствующих атрибутов. В теле таблицы одна строка соответствует одной сущности, представленной своим условным знаком (*e1*, *e5*) и значениями атрибутов.

Шапка таблицы для множеств связей – трехуровневая. На верхнем уровне представлены роли сущностей в связях (количество начальных столбцов для сущностей и их ролей соответствует степени множества связей). На втором уровне слева указано имя множества связей, далее следуют имена атрибутов (количество правых столбцов определяется количеством атрибутов у множества связей). Третий уровень шапки предназначен для задания имен множеств сущностей, участвующих во множестве связей в соответствующих ролях, и множеств значений, являющихся ОЗО атрибутных отображений. Строки тела таблицы соответствуют самим связям. В них указаны условные знаки сущностей и значения атрибутов.

3.2.2. Ограничения целостности

1) Ограничения целостности на значения атрибутов

Множество допустимых значений атрибутов можно задавать:

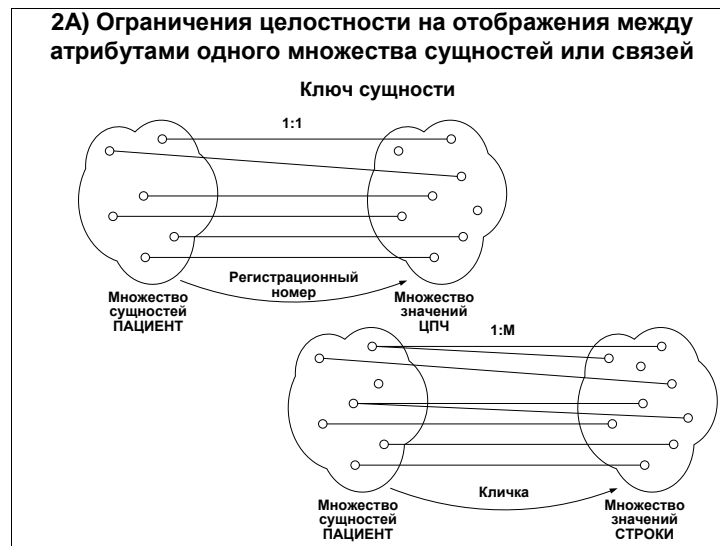
- 1) указанием соответствующего множества значений
 Фамилия: ПАЦИЕНТ -> Строки
 Номер койки: РАЗМЕЩЕНИЕ -> ЦПЧ
- 2) сравнением с константой или значением атрибута
 (возможны более сложные выражения)
 Назначенное время (a) < 24, где a ∈ АНАЛИЗ
- 3) перечислением значений
 Пол (a) ∈ {'м', 'ж'}, где a ∈ ПАЦИЕНТ
- 4) более сложным логическим выражением, включающим
 в виде атомов конструкции 2 – 3
 Назначенное время (a) ≥ 0 AND
 Назначенное время (a) < 24, где a ∈ АНАЛИЗ

Средства определения ОЦ в ER-модели, по сути, не выходят за рамки тех возможностей фиксации закономерностей ПрО, которые мы рассмотрели во второй главе. Здесь представлен материал, почерпнутый из самых различных источников, в том числе и основополагающей статьи Чена. Следует отметить, что отсутствующие инструменты ОЦ можно легко добавить, зная основные принципы и не нарушая сложившейся в ER-модели терминологии и нотации.

Что касается ограничений целостности на значения атрибутов, то их задание в ER-модели достаточно традиционно и осуществляется в текстовой нотации. При этом для атрибутивных отображений используется функциональная форма, например:

$$\forall e \in \text{ЧЕЛОВЕК} (\text{Рост}(e) \geq 50 \text{ AND } \text{Рост}(e) \leq 300).$$

Как видим, для декларации ОЦ на значения атрибутов предлагается использовать язык исчисления предикатов первого порядка с переменными, пробегающими по множествам сущностей или связей и принимающими значения, соответственно, сущностей или связей. Если квантор явно не указывается, предполагается квантор всеобщности («∀»). Областью действия ОЦ первого типа, как правило, является одно множество сущностей или одно множество связей.



Атрибут может использоваться для уникальной идентификации сущностей во множестве сущностей. Возможно, для идентификации сущностей понадобится более одного атрибута.

Определение 3.2.6. Ключ сущности (англ. entity key) – это группа атрибутов, такая, что отношение между множеством сущностей и Декартовым произведением соответствующих множеств значений есть отношение типа 1:1 или 1:M (для многозначных атрибутов). В случае, когда ключ составляет один атрибут, вместо Декартова произведения рассматривается одиночное множество значений. Таким образом, в обоих случаях требуется функциональность отображения во множество сущностей.

На слайде показаны ситуации с обоими типами отношений для простых одноатрибутных ключей. Первый вариант определяет в качестве ключа однозначный атрибут множества сущностей *ПАЦИЕНТ* *Регистрационный номер*, второй – многозначный атрибут *Кличка*. В обоих случаях предполагается функциональность отображения из множества значений в множество сущностей.

В языке ОЦ объявление ключа сущности осуществляется описателем *KEY*, сопоставляемым атрибуту или группе атрибутов множества сущностей. В текстовой нотации определения схемы множества сущностей ключи могут подчеркиваться.

Отметим, что наличие ключа не является обязательным требованием к множеству сущностей. Среди естественных атрибутов и их групп может и не оказаться ключей. С другой стороны, у одного множества сущностей может быть несколько ключей.

Особого смысла, выделять среди многочисленных ключей сущности первичный ключ, нет, хотя иногда это понятие вводят в ER-модель. Наличие у сущностей уникальных указателей, с помощью которых происходит идентификация сущностей в БД, и реализуются связи между ними, приводит к тому, что теряется основное назначение первичных ключей – идентификация объектов БД при их взаимодействии между собой.

В ER-модели существует еще одна любопытная возможность идентификации сущностей, когда сущность не может быть идентифицирована исключительно по значениям своих собственных атрибутов, но может быть идентифицирована с участием ее связи с другой сущностью. Об этом мы поговорим чуть далее.

В публикациях по ER-модели абсолютно не упоминаются ОЦ, связанные с недопустимостью неопределенных значений атрибутов. При желании этот пробел можно восполнить введением соответствующих описателей.



Определение 3.2.7. Множества связей также могут иметь **ключи связей** (англ. relationship key). Возможны три способа уникальной идентификации связей:

- 1) только по сущностям, в ней участвующим (одной, нескольким или всем);
- 2) по сущностям, в ней участвующим, и значениям одного или нескольких атрибутов множества связей;
- 3) только по значениям одного или нескольких атрибутов множества связей.

Первая ситуация показана на примере множества связей *ВРАЧ-ПАЦИЕНТ* (для идентификации связи требуется указать сущности обоих множеств сущностей) и множества связей *АНАЛИЗ ПАЦИЕНТА* (для идентификации связи достаточно сущности типа *АНАЛИЗ*). Как вы, наверное, догадываетесь, последнюю возможность обеспечивает функциональность отображения из множества сущностей *АНАЛИЗ* во множество сущностей *ПАЦИЕНТ* в контексте множества связей *АНАЛИЗ ПАЦИЕНТА*. Для множеств связей степени больше двух возможна ситуация, при которой для уникальной идентификации связи достаточно указать несколько, но не все сущности.

В ситуации с множеством связей *РАЗМЕЩЕНИЕ* можно в качестве ключа использовать сущность типа *ПАЦИЕНТ*. Другим ключом этого множества связей будет комбинация сущности типа *ПАЛАТА* и значения атрибута множества связей *Номер койки* (второй способ идентификации связей).

До сих пор мы предполагали, что связи типа *БРАК* определяют взаимнооднозначное соответствие между мужчинами и женщинами (бинарное отношение типа 1:1). Это предположение строилось в условиях фиксации в БД только текущего состояния этих связей. Если говорить о БД бюро ЗАГС, то в ней необходимо хранить всю историю браков. В таком случае у одного человека (мужчины или женщины) может быть много жен или мужей (естественно, в разные моменты времени), и отношение становится отношением типа *M:N*.

В изменившихся семантических условиях ключом множества *БРАК* может быть комбинация из одной сущности (безразлично в какой роли – *Муж* или *Жена*) и значения атрибута множества связей *Дата регистрации брака*. По-видимому, зарегистрировать два и более брака в один день одному человеку достаточно сложно. Но ведь возможна уникальная идентификация связи и по значению атрибута *Серия и номер свидетельства* (можно рассматривать в качестве ключа комбинацию двух атрибутов *Серия свидетельства* и *Номер свидетельства*). Последняя ситуация является примером третьего способа идентификации связей.



Бинарное множество связей, как и любое отношение, определяет пару отображений между множествами сущностей. Каждое из этих отображений может быть, а может и не быть полностью определенным и функциональным. Полная определенность отображения говорит о наличии зависимости существования, сочетания функциональных и нефункциональных свойств обоих отображений определяют тип бинарного множества связей – $1:1$, $1:M$, $M:1$, $M:N$.

Как поступать в случае множества связей степени больше двух, в публикациях по ER-модели умалчивается. Будучи вооруженными понятием сложного отображения, мы можем проводить анализ ОЦ и в этом случае. Рассмотрим его на примере тернарного множества связей *НАПРАВЛЕНИЕ НА АНАЛИЗ*. Это множество связей определяет следующие отображения между множествами сущностей:

- 1) *ВРАЧ* -> *АНАЛИЗ* × *ПАЦИЕНТ* с кардинальными числами $(0, \infty)$;
- 2) *АНАЛИЗ* × *ПАЦИЕНТ* -> *ВРАЧ* с кардинальными числами $(0, 1)$;
- 3) *АНАЛИЗ* -> *ВРАЧ* × *ПАЦИЕНТ* с кардинальными числами $(1, 1)$;
- 4) *ВРАЧ* × *ПАЦИЕНТ* -> *АНАЛИЗ* с кардинальными числами $(0, \infty)$;
- 5) *ПАЦИЕНТ* -> *АНАЛИЗ* × *ВРАЧ* с кардинальными числами $(0, \infty)$;
- 6) *АНАЛИЗ* × *ВРАЧ* -> *ПАЦИЕНТ* с кардинальными числами $(0, 1)$.

Некоторые из кардинальных чисел требуют пояснения. МинКЧ, равное 0 у 2-го и 6-го отображений, объясняется тем, что в кортеже прообраза этих отображений могут оказаться анализ и пациент или врач, не имеющие к этому анализу никакого отношения. Если же подберется соответствующая пара, то она породит не более одного образа. Каждая сущность типа *АНАЛИЗ*, естественно, соответствует одной и только одной паре типа «врач-пациент».

Определение 3.2.8. ER-модель позволяет явно представить ограничение по существованию сущностей (называемое в ER-модели **зависимостью существования** – E-зависимостью). Зависимое множество сущностей называется **множеством слабых сущностей** (англ. weak entity set), а ассоциированное с ним множество связей – **множеством слабых связей** (англ. weak relationship set). Множества сущностей, не принадлежащие к этому виду, носят название **множеств регулярных сущностей** (англ. regular entity set). Множества связей, в контексте которых не порождаются множества слабых сущностей, называются **множествами регулярных связей** (англ. regular relationship set).

В нашей демонстрационной ПрО наблюдаются следующие полностью определенные отображения между множествами сущностей:

- *ПАЛАТА* -> *БОЛЬНИЦА*;
- *ДИАГНОЗ* -> *ПАЦИЕНТ*;

- *АНАЛИЗ* -> *ПАЦИЕНТ*;
- *АНАЛИЗ* -> *ЛАБОРАТОРИЯ*.

ООО этих отображений являются соответственно множествами слабых сущностей:

- *ПАЛАТА* в контексте множества слабых связей *БОЛЬНИЧНАЯ ПАЛАТА*;
- *ДИАГНОЗ* в контексте множества слабых связей *ДИАГНОЗ ПАЦИЕНТА*;
- *АНАЛИЗ* в контексте множества слабых связей *АНАЛИЗ ПАЦИЕНТА*;
- *АНАЛИЗ* в контексте множества слабых связей *ОБРАБОТКА АНАЛИЗА*.

Имейте в виду, что в контексте различных множеств связей множество сущностей может оказаться как множеством слабых сущностей, так и множеством регулярных сущностей. Так множество сущностей *ПАЛАТА* в контексте множества регулярных связей *ПЕРСОНАЛ ПАЛАТЫ* является множеством регулярных сущностей.

Множества слабых связей могут быть любого вида – бинарные или n -арные, $1:1$, $1:M$ или $M:N$. В последнем случае удаление регулярной сущности не обязательно влечет за собой удаление связанных с ней слабых сущностей, так как последние могут быть связаны с другими регулярными сущностями.

Примером множества слабых связей степени три является множество связей *НАПРАВЛЕНИЕ НА АНАЛИЗ*. Полное функциональное отображение *АНАЛИЗ* -> *ВРАЧ* \times *ПАЦИЕНТ* говорит о слабости сущностей типа *АНАЛИЗ* в контексте этого множества связей.

Бинарное множество связей *РОЖДЕНИЕ* также является множеством слабых связей, хотя и имеет тип $M:N$. Действительно, отображение *РЕБЕНОК* -> *РОДИТЕЛЬ* является полностью определенным, а значит множество сущностей *ЧЕЛОВЕК* в роли *Ребенок* – множество слабых сущностей.

Определение 3.2.9. Часто на практике используется частный случай зависимости существования – **зависимость по идентификации** (ID-зависимость). При этом сущности одного (зависимого) множества сущностей не просто обязаны иметь связи с сущностями другого множества сущностей (возможно, того же самого в случае рекурсивных связей), но еще, и не могут быть идентифицированы без них. Этот случай является еще одним способом уникальной идентификации сущностей. В нашей демонстрационной ПрО есть показательный пример ID-зависимости. Множество сущностей *ПАЛАТА* является ID-зависимым от множества сущностей *БОЛЬНИЦА* в контексте множества связей *БОЛЬНИЧНАЯ ПАЛАТА*. Действительно, ключей среди атрибутов множества сущностей *ПАЛАТА* нет, но сущности этого множества могут быть уникально идентифицированы комбинацией связанной с ней сущности типа *БОЛЬНИЦА* и значения атрибута множества сущностей *ПАЛАТА* *Номер палаты*.

Еще раз отметим, что ID-зависимость одновременно является зависимостью существования. Обратное, однако, не обязательно.

Ограничения целостности типа 2Б могут быть представлены на ER-диаграмме. Максимальные кардинальные числа отображений, определяющие тип отношения, задаваемого множеством связей, указываются в виде дополнительных пометок на ребрах ролей. Буквенная метка (обычно используются латинские буквы M , N , P , ...) указывает, что максимальное кардинальное число не ограничено (равно ∞).

В случае бинарного множества связей вопросов с пометками не возникает. Они соответствуют типу этого множества связей. В случае множества связей степени больше двух одной пометки на ребре уже недостаточно. Для тернарного случая по две пометки на разных концах ребер достаточно для характеристики МаксКЧ всех отображений между множествами сущностей, определяемых множеством связей. Для множеств связей степени больше 3-х можно рекомендовать делать также по две пометки на ребре: одна (возле множества связей) для отображения, определяемого ролью, другая (ближе к множеству сущностей) – для отображения, определяющего роль. Когда желают ограничиться одной пометкой на ребре, обычно указывают МаксКЧ отображения, определяющего роль. Именно так мы поступаем в бинарном случае.

Покажем, как это осуществляется на примере роли *Врач* множества связей *НАПРАВЛЕНИЕ НА АНАЛИЗ*. Отображение *АНАЛИЗ* × *ПАЦИЕНТ* → *ВРАЧ* (в параграфе 3.3 мы назовем такие отображения отображениями, определяющими роль) функционально, поэтому на ребре роли рядом с множеством сущностей *ВРАЧ* ставим пометку «1». Обратное ему отображение *ВРАЧ* → *АНАЛИЗ* × *ПАЦИЕНТ* (определяемое ролью) нефункционально, поэтому на том же ребре рядом с множеством связей ставим пометку «М».

Наличие ограничения по существованию, наложенного на определенное множество сущностей, показывают в ER-диаграмме очерченным двойными линиями прямоугольником, изображающим множество слабых сущностей, меткой *E*, проставляемой в ромбе, изображающем соответствующее множество слабых связей, и стрелкой, указывающей на зависимое множество сущностей. ID-зависимость представляется аналогично ограничению по существованию, за исключением того, что в ромбе, изображающем соответствующее множество связей, проставляется метка *ID*.

Для множеств связей степени больше двух в этом случае ничего не меняется. Так, полная определенность отображения *АНАЛИЗ* → *ВРАЧ* × *ПАЦИЕНТ* приводит к слабости сущностей типа *АНАЛИЗ* и соответствующим обозначениям в ER-диаграмме.



На слайде представлена полная ER-диаграмма для нашей медицинской ПрО с учетом ограничений целостности.

Эта ER-диаграмма вместе с приведенными ниже определениями атрибутов множеств сущностей и множеств связей составляет ER-схему этой ПрО, которой мы в дальнейшем будем в основном пользоваться.

БОЛЬНИЦА (Название, Адрес, Телефон (М), Число коек (Ч/К))

ПАЛАТА (Номер палаты (Н/П), Название, Число коек (Ч/К))

ПЕРСОНАЛ (Фамилия, Должность, Смена, Зарплата (З/П))

ВРАЧ (Фамилия, Специальность)

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Адрес, Дата рождения (Д/Р), Пол, Номер медицинского полиса (НМП))

ДИАГНОЗ (Тип диагноза (Т/Д), Осложнения, Предупреждающая информация)

ЛАБОРАТОРИЯ (Название, Адрес, Телефон (М))

АНАЛИЗ (Тип анализа (Т/А), Назначенная дата (Н/Д), Назначенное время (Н/В), Номер направления (Н/Н), Состояние)

РАЗМЕЩЕНИЕ (Номер койки (Н/К))

Следует иметь в виду, что здесь нашли отражение отнюдь не все элементы схемы, в частности, отсутствуют очевидные ОЦ на значения атрибутов.

3.2.3. Операции

Операции навигационного языка	
1)	Создание сущностей (с возможностью задания экземпляров атрибутивных отображений)
2)	Изменение сущностей (изменение экземпляров атрибутивных отображений)
3)	Удаление сущностей
4)	Создание связей (с указанием сущностей для всех ролей и экземпляров атрибутивных отображений)
5)	Изменение связей (изменение экземпляров атрибутивных отображений)
6)	Удаление связей
7)	Селекция сущности по указателю текущей
8)	Селекция связи по указателю текущей
9)	Селекция сущности по указателю текущей и по данным
10)	Селекция связи по указателю текущей и по данным
11)	Селекция сущности по указателю текущей и по связи
12)	Селекция связи по указателю текущей и по связи
13)	Селекция сущности по указателю текущей, по данным и по связи
14)	Селекция связи по указателю текущей, по данным и по связи
15)	Выборка значений атрибутов для текущей сущности или связи

Как уже отмечалось, операции над данными – наименее актуальный компонент в любой семантической модели и в ER-модели, в частности. Это связано с тем, что применяются эти модели в основном при проектировании схемы БД, когда о манипулировании данными речь не идет. Ни одна из семантических моделей не реализована ни в одной СУБД, и поэтому БД в них не создаются. Хотя, как мы увидим в следующем параграфе, операции над данными могут применяться и в таких условиях. Но об этом позже.

Первоначальные публикации по ER-модели не содержали спецификаций языка данных, хотя и указывалось, что информационные запросы могут быть выражены в понятиях операций над множествами.

Мы попробуем сами синтезировать навигационный язык для манипулирования данными в ER-модели Чена.

Основными элементами данных в ней являются сущности и связи, организованные в форме множеств. Значения в ER-модели самостоятельной ценности не имеют и используются исключительно в контексте сущностей и связей. Поэтому в языке достаточно предусмотреть понятия указателей текущих сущностей и связей, пробегающих по соответствующим множествам. Причем необходимо предоставить возможность неоднократного одновременного их определения для обеспечения параллельных независимых сканирований одних и тех же множеств.

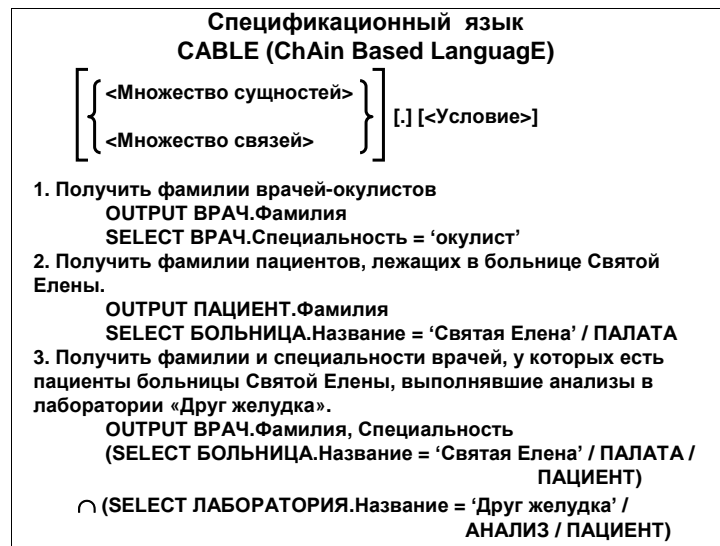
Если взять за основу граф экстенционала БД, то очевидно, что в навигационном языке необходимо обеспечить операции для создания и удаления всех элементов (вершин и дуг) такого графа. Таким образом в языке появились операции 1 – 6, представленные на слайде.

Для обеспечения обхода графа знаков необходимо предусмотреть в языке полный набор операций, осуществляющих установку указателей текущих для основных элементов данных. При этом надо использовать все виды селекции с обязательным включением в каждую такую операцию селекции по указателю текущей. Получаем операции 7 – 14, обеспечивающие указанную функциональность.

По-видимому, селекция по текущей, осуществляющая последовательный проход в прямом и обратном направлении по элементам одного множества, в пояснении не нуждается. Селекция по данным предполагает дополнительную фильтрацию перебираемых элементов по некоторому условию, включающему сравнения значений атрибутов этих элементов с константами. Селекция по связям в данном случае позволяет

осуществлять переходы от сущности к связи, в которой она участвует, или от связи к сущности, в ней участвующей.

И, наконец, последняя операция выборки предназначена для получения значений атрибутов текущей сущности или связи.



Одним из спецификационных языков ER-модели является язык CABLE – ChAin-Based Language – язык, основанный на цепочках.

Основной акцент при построении языка сделан на операциях выборки данных, причем существенно используется концепция цепочек или путей, включающих множества сущностей и множества связей БД. Цепочка или путь в БД определяется элементами запроса, называемыми бусинками. Каждая бусинка задает точку маршрута и/или элементарный критерий селекции сущностей или связей по данным в этой точке маршрута (ее грамматика приведена в верхней части слайда).

Все компоненты структуры бусинки являются необязательными. Если имена атрибутов, указанные в условии, уникальны в схеме, то имя множества сущностей или связей может быть опущено. Если в бусинке не задано условие, то указанное множество необходимо для однозначной идентификации пути в БД. Бусинка пути может быть вообще опущена, если нет необходимости в селекции по данным, и проблема выбора пути не стоит.

При переходе от одной точки маршрута к следующей осуществляется селекция элементов очередного множества, имеющих связи с элементами, селектированными на предыдущем шаге (селекция по связям). Если в бусинке очередной точки маршрута указано условие селекции по данным, оно применяется к полученному селекцией по связям подмножеству, оставляя в нем только те элементы, которые удовлетворяют указанному условию.

В запросе может быть специфицирован не только линейный маршрут от исходного множества к целевому, но и дерево. В последнем случае листья задают исходные множества, а корень дерева – целевое множество. В вершинах, в которых происходит слияние двух и более путей, осуществляются теоретико-множественные операции объединения, пересечения или разности подмножеств, полученных при прохождении по каждому из соединяющихся маршрутов.

На слайде показаны три запроса на языке CABLE. Граф переходов первого запроса состоит из одной вершины, представляющей множество сущностей *ВРАЧ*. Действительно, это множество является и исходным, и целевым одновременно. Выполняя этот запрос, система осуществит селекцию сущностей типа *ВРАЧ* по указанному критерию и выдаст для полученных сущностей значения атрибута *Фамилия*.

Граф второго запроса – маршрут, состоящий из следующих точек:

- 1) множество сущностей *БОЛЬНИЦА* – исходное множество, в котором осуществляется первоначальная селекция по данным;

- 2) множество связей *БОЛЬНИЧНАЯ ПАЛАТА* – точка маршрута, бусинка для которой не указана, поскольку между множеством сущностей *БОЛЬНИЦА* и множеством сущностей *ПАЛАТА* имеется только одно множество связей;
- 3) множество сущностей *ПАЛАТА* («/» – знак, отделяющий одну бусинку запроса от другой);
- 4) множество связей *РАЗМЕЩЕНИЕ* – опять точка маршрута без бусинки;
- 5) множество сущностей *ПАЦИЕНТ* – целевое множество.

Граф третьего запроса – дерево с двумя ветвями. Маршрут первой ветви: *БОЛЬНИЦА* -> *БОЛЬНИЧНАЯ ПАЛАТА* -> *ПАЛАТА* -> *РАЗМЕЩЕНИЕ* -> *ПАЦИЕНТ*. Маршрут второй ветви: *ЛАБОРАТОРИЯ* -> *ОБРАБОТКА АНАЛИЗА* -> *АНАЛИЗ* -> *АНАЛИЗ ПАЦИЕНТА* -> *ПАЦИЕНТ*. В точке слияния этих маршрутов (множестве сущностей *ПАЦИЕНТ*) будет произведена указанная операция пересечения полученных подмножеств, и таким образом сформировано подмножество пациентов, для которых одновременно удовлетворяются оба условия запроса. После этого остается осуществить лишь переход к целевому множеству по следующему маршруту: *ПАЦИЕНТ* -> *ВРАЧ-ПАЦИЕНТ* -> *ВРАЧ*.

3.2.4. Назначение ER-модели

Критерии качества семантической модели

1. Модель должна обладать достаточной общностью и ясностью для того, чтобы в ней можно было легко представить любые явления и законы моделируемого мира.
2. Разрыв между этой моделью и моделями, реализуемыми в СУБД, не должен быть большим, желательно, чтобы можно было воспользоваться максимально формальными правилами преобразования структур и ограничений целостности из одной модели в другую.

Появление модели «Сущность-Связь» обусловлено потребностями процесса проектирования БД. Модель создавалась с учетом двух основных критериев.

1. Модель должна обладать достаточной общностью и ясностью для того, чтобы в ней можно было легко представить любые явления и законы моделируемого мира.

2. Разрыв между этой моделью и моделями, реализуемыми в СУБД, не должен быть большим, желательно, чтобы можно было воспользоваться максимально формальными правилами преобразования структур и ограничений целостности из одной модели в другую.

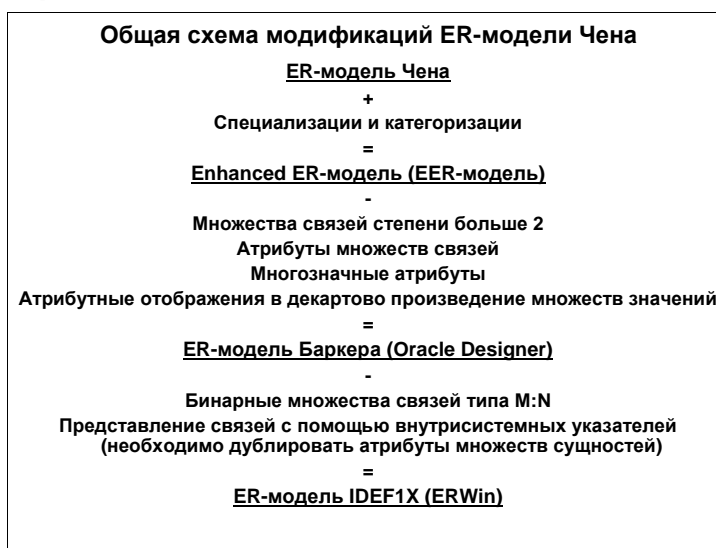
Модели вида «Сущность-Связь» отвечают задачам анализа и проектирования в плане представления исходных данных. Они помогают аналитикам контактировать как с пользователями, так и с проектировщиками информационной системы в процессе анализа требований и ее конструирования.

Вообще же целью исследований в области семантического моделирования является совершенствование семантической методики проектирования схем БД и доведение ее до идеала, при котором выявление семантики ПрО и полная ее формализация осуществляется человеком один раз на первом этапе разработки информационной системы – анализе ПрО. Дальнейший перевод закономерностей ПрО, представленных в семантической схеме, на язык СУБД может происходить в достаточной степени формально одним из следующих способов:

- вручную человеком по фиксированным правилам семантической методики;
- полностью автоматически с использованием CASE-инструмента;
- в ходе человеко-машинной процедуры под управлением CASE-системы.

Для достижения поставленной цели как минимум необходимо синтезировать мощную семантическую модель и разработать для нее наборы детальных правил трансформации схемы в СУБД-ориентированные модели данных. Программой-максимумом является реализация CASE-системы, позволяющей вводить в удобной для человека форме семантику ПрО и генерировать элементы схемы на языке конкретной СУБД.

3.2.5. Модификации ER-модели Чена

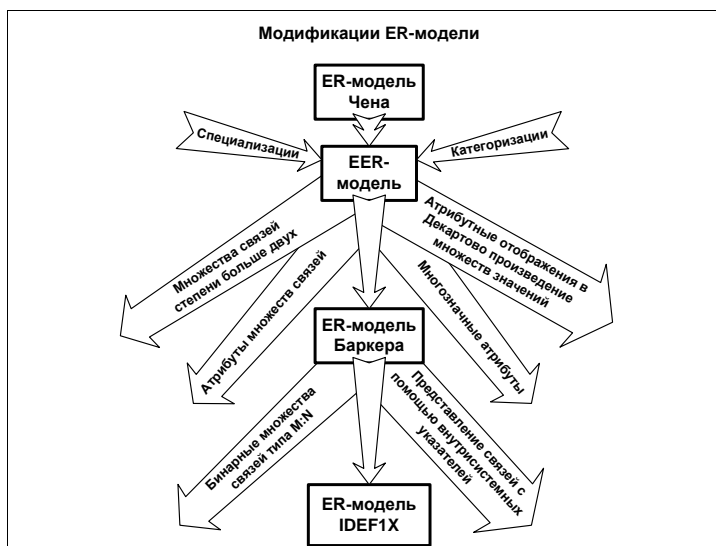


После первых публикаций по ER-модели в конце 70-х годов XX века наблюдался взрыв исследовательского интереса к семантическому моделированию. Регулярно проводились международные конференции, печатались статьи и книги, посвященные развитию семантических моделей и ER-модели, в частности.

Мы познакомимся только с некоторыми модификациями ER-модели Чена, сейчас широко известными и используемыми на практике, – расширенной ER-моделью, ER-моделью Баркера и ER-моделью IDEF1X. На этом и последующем слайде представлена общая схема осуществления этих модификаций.

За исходную основу взяты структурные понятия ER-модели Чена. Лишь EER-модель привнесла новые выразительные возможности, не нарушив полноту и простоту восприятия концепций ER-модели Чена. Каждая последующая модификация нещадно отбрасывала те или иные концепции, ухудшая тем самым способности модели по выражению семантики Про.

Можно сказать, что развитие ER-модели шло по пути резкой потери качества в смысле первого критерия семантической модели (пункт 3.2.4) и все лучшего удовлетворения второго критерия. Дело дошло до того, что основные концепции ER-модели IDEF1X уже практически ничем не отличаются от моделей данных реляционных СУБД. Такую модель можно назвать семантической уже с большой натяжкой.



Итак, ER-модель была пополнена понятиями «специализация» и «категоризация» для представления отношений обобщения между множествами сущностей, и, таким образом, образована расширенная ER-модель (Enhanced Entity-Relationship Model – EER-модель).

Последующие модификации ER-модели (нотации Баркера и IDEF1X) хоть и удостоились реализаций в CASE-инструментах (Oracle Designer и ERwin соответственно) шли по пути сокращения выразительных возможностей представления семантики предметной области.

Так, в ER-модели Баркера уже не удастся напрямую использовать такие естественные для человеческого восприятия формы, как:

- множества связей степени больше двух,
- атрибуты множеств связей,
- многочисленные атрибуты,
- атрибутные отображения в Декартово произведение множеств значений.

Нотация IDEF1X также лишена этих средств выражения семантики ПрО. Но ее авторы на этом не остановились и с удовольствием продолжили процесс низведения ER-модели до уровня реляционной модели данных. Они запретили использовать в окончательной схеме даже бинарные множества связей типа $M:N$. Их надо представлять в виде дополнительного множества сущностей и пары множеств связей типа $1:M$. Но самое ужасное, что в нотации IDEF1X проектировщик вынужден отказаться от внутрисистемных указателей сущностей и в явной форме оперировать первичными и внешними ключами. Это значит, что он сам (а не CASE-инструмент) должен заботиться о создании суррогатных ключей (ведь множество сущностей не обязано иметь ключ среди своих естественных атрибутов), безошибочном дублировании атрибутов для представления связей и принятии решения о реализации зависимостей существования и идентификации. Ну, чем не реляционная модель!

Основным недостатком ER-моделей Баркера и IDEF1X является тот факт, что они предлагают бедный язык для описания семантики ПрО и, в силу этого, уже на начальной фазе формализации представлений о ПрО вынуждают человека принимать «чисто технические» решения. Но совмещение двух в принципе различных мыслительных процессов – выяснение семантики ПрО и определение схемы БД на весьма абстрактном языке СУБД – еще полбеда. Хуже то, что принимаемые в ходе этого технические решения (например, введение дополнительного множества сущностей и бинарных множеств связей вместо множества связей степени больше двух) должны быть по-хорошему проанализированы заново в реляционной схеме. В противном случае можно получить не самую эффективную схему БД.

Специализации и категоризации

1. Специализации и категоризации являются традиционной естественной формой представления информации для человека.
2. Специализация предполагает перенос определений некоторых атрибутов и множеств связей на уровень суперкласса, благодаря этому фиксируется важная информация о том, что данные атрибуты и типы связей имеют смысл не только для экземпляров конкретного подкласса, но и для суперкласса в целом, а, значит, и для всех остальных его подклассов.
3. Специализации и категоризации позволяют исключить дублирующие определения атрибутов и множеств связей для «родственных» множеств сущностей, в результате чего удается избежать некоторых ошибок и сократить время, затрачиваемое на проектирование.
4. Процесс семантического моделирования зачастую ведется в терминах различных абстракций одного и того же объекта (например, конкретный человек может быть представлен и сущностью типа ВРАЧ, и сущностью типа ПАЦИЕНТ одновременно), а специализации и категоризации позволяют в таких случаях восстановить целостность объекта реального мира.
5. Выделение подкласса в процессе специализации является своего рода гарантией применимости связей и значений атрибутов только к тем сущностям, для которых они актуальны.
6. Специализация призвана упростить процесс моделирования, позволяя постепенно увеличивать сложность схемы, в то время как генерализация предоставляет проектировщику возможность работать с большими схемами, выбирая различные уровни абстракции.

Как уже отмечалось, расширениями ER-модели Чена являются концепции специализации и категоризации. С их помощью задаются отношения типа «подкласс-суперкласс» между множествами сущностей. Суперклассы и подклассы используются с целью исключения дублирования определения общих атрибутов и типов связей нескольких «родственных» типов сущностей. Такие определения выносятся на уровень суперкласса и наследуются (англ. *inherit*) всеми его подклассами. Подкласс также является множеством сущностей, а потому может иметь свои собственные подклассы.

Определение 3.2.10. Специализация (англ. *specialization*) представляет собой нисходящий подход к определению множества суперклассов и связанных с ними подклассов.

Определение 3.2.11. Генерализация (англ. *generalization*) представляет собой восходящий подход (противоположный специализации), который позволяет создать обобщенный суперкласс на основе различных исходных подклассов.

Множество подклассов определяется на основе некоторых отличительных характеристик подмножеств сущностей суперкласса – специфических значений атрибутов и связей. Для одного и того же множества сущностей можно выделить несколько независимых специализаций, основываясь на разных его характеристиках. Возможно использование множественного наследования (англ. *multiple inheritance*), когда некоторому подклассу соответствует сразу несколько суперклассов (естественно, в разных специализациях).

Каждый объект ПрО, представленный сущностью подкласса, должен обязательно иметь и сущность суперкласса специализации. Обратное, однако, не обязательно.

Определение 3.2.12. В некоторых ситуациях может потребоваться смоделировать связь «суперкласс/подкласс», включающую сразу несколько разных суперклассов, и каждая сущность подкласса должна соответствовать лишь одной сущности одного из суперклассов. В этом случае создаваемый подкласс будет называться **категорией** (англ. *category*), а процесс порождения категории, основанной на нескольких суперклассах, называется **категоризацией** (англ. *categorization*). Подкласс категории обладает выборочным наследованием (англ. *selective inheritance*). Это означает, что каждая сущность категории наследует атрибуты и типы связей только одного суперкласса. Поэтому не надо путать категоризацию с множественным наследованием, при котором осуществляется наследование от всех суперклассов. Каждый объект ПрО, представленный сущностью в категории, должен иметь сущность одного и только одного суперкласса категоризации.

Кроме уже указанных возможностей, обеспечивающихся механизмом наследования, специализации и категоризации позволяют восстановить в БД единство

объекта предметной области, рассматриваемого в противном случае как несвязанные его абстракции-сущности.

Ограничения целостности на специализации и категоризации

На специализации могут быть наложены ограничения.

Первое ограничение называется **ограничением непересечения**. Оно гласит, что если подклассы некоторой специализации не пересекаются, то каждый объект может быть представлен сущностью только одного из подклассов данной специализации. Если подклассы специализации пересекаются, один объект может быть представлен сущностями сразу нескольких подклассов специализации.

Второе ограничение специализации называется **ограничением участия**, оно может быть полным или частичным. Специализация с полным участием означает, что каждый объект, представленный сущностью суперкласса, должен быть представлен сущностью хотя бы одного подкласса этой специализации. Специализация с частичным участием означает, что объект, представленный сущностью суперкласса, не обязательно должен быть представлен сущностью какого-либо подкласса этой специализации.

Категоризация может быть дополнительно детализирована с учетом **полного или частичного участия** объектов суперклассов в категории. При полном участии каждый объект всех суперклассов должен быть представлен сущностью данной категории. При частичном участии это ограничение устраняется, и всем объектам всех суперклассов иметь сущности данной категории не обязательно.

Определение 3.2.13. На специализации также могут быть наложены ограничения. Первое ограничение называется **ограничением непересечения** (англ. disjoint constraint). Оно гласит, что если подклассы некоторой специализации не пересекаются, то каждый объект может быть представлен сущностью только одного из подклассов данной специализации. Если подклассы специализации пересекаются, один объект может быть представлен сущностями сразу нескольких подклассов специализации.

Определение 3.2.14. Второе ограничение специализации называется **ограничением участия** (англ. participation constraint), оно может быть полным или частичным. Специализация с полным участием означает, что каждый объект, представленный сущностью суперкласса, должен быть представлен сущностью хотя бы одного подкласса этой специализации. Специализация с частичным участием означает, что объект, представленный сущностью суперкласса, не обязательно должен быть представлен сущностью какого-либо подкласса этой специализации.

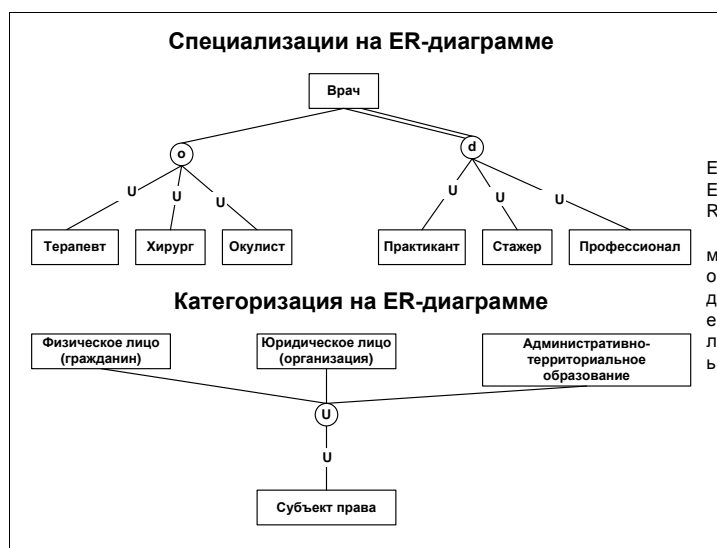
Как и операция специализации, операция категоризации может быть дополнительно детализирована с учетом полного или частичного участия объектов суперклассов в категории. При полном участии каждый объект всех суперклассов должен быть представлен сущностью данной категории. При частичном участии это ограничение устраняется, и всем объектам всех суперклассов иметь сущности данной категории не обязательно.

Итак, подклассы специализаций могут пересекаться или не пересекаться, а участие может быть полным или частичным. Эти свойства специализаций независимы, поэтому на практике могут встретиться специализации с четырьмя различными комбинациями свойств:

- полные, пересекающиеся специализации;
- частичные, пересекающиеся специализации;
- полные, непересекающиеся специализации;
- частичные, непересекающиеся специализации.

Аналогично, возможны два варианта категоризаций:

- полные категоризации;
- частичные категоризации.



Примеры двух специализаций и одной категоризации показаны на слайде в виде фрагментов ER-диаграмм.

Подклассы специализации соединяются линиями с кружком, который, в свою очередь, соединяется с суперклассом. Символ включения множества в другое множество (\subset) на каждой линии, соединяющей подкласс с кружком, указывает на подкласс. Общие для всех подклассов атрибуты и множества связей соединяются с прямоугольником, обозначающим суперкласс. Специфические для каждого подкласса атрибуты и множества связей соединяются с прямоугольником, обозначающим этот подкласс.

Линия, соединяющая подкласс-категорию с кружком категоризации, помечается символом включения множества в другое множество (\subset), а в кружок категоризации помещается символ объединения (\cup).

Для представления непересекающейся (англ. **disjoint**) специализации используется символ «d», который располагается в центре кружка специализации. Для представления пересекающейся (англ. **overlapping**) специализации используется символ «o». Для обозначения полного участия между суперклассом и кружком специализации проводят двойную линию, для частичного участия – одинарную.

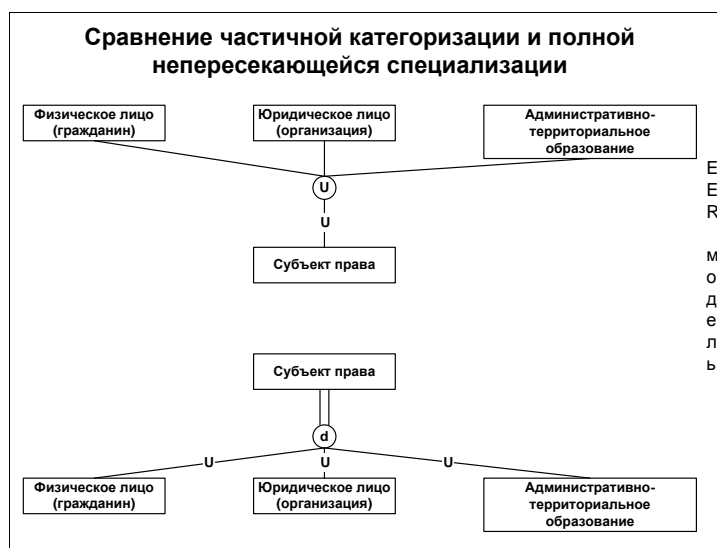
Аналогично на ER-диаграмме показывают полное и частичное участие сущностей суперклассов в подклассе категории, только двойной или одинарной делают в этом случае линию, соединяющую подкласс с кружком категоризации.

В верхней части слайда показаны две независимые специализации врачей. Слева – частичная пересекающаяся специализация врачей на основании наличия у них определенных специальностей, подтвержденного соответствующими лицензиями. Поскольку в специализации указаны явно не все виды специальностей (предполагается, что особый интерес представляют именно эти), специализация является частичной (врачи, для которых не нашлось места в подклассах, будут представлены сущностью только суперкласса – множества сущностей *ВРАЧ*). Пересекающейся эта специализация является потому, что наверняка найдутся врачи, работающие по нескольким специальностям.

Вторая специализация врачей по своим свойствам диаметрально противоположна первой, она – полная и непересекающаяся. Это произошло потому, что в этом случае основанием деления врачей по указанным подклассам является документально подтвержденный уровень квалификации врача. Практиканты еще не получили разрешения на самостоятельное лечение пациентов. Стажеры имеют такое разрешение, но набирают необходимый стаж самостоятельной работы. Профессионалы работают с пациентами без каких-либо организационных ограничений.

Категоризации встречаются в отличие от специализаций гораздо реже. В нижней части слайда представлена частичная категоризация. В ней отражен тот факт, что по

Гражданскому кодексу РФ субъекты права составляют граждане, организации и административно-территориальные образования. Эта категоризация неполна потому, что, по крайней мере, не все граждане вступили в правовые отношения и не могут считаться субъектами права.



Со специализациями человек постоянно имеет дело в повседневной жизни, поэтому их концепция ему понятна и не вызывает трудностей в использовании. Понять суть категоризации сложнее, но желательно это тоже сделать, поскольку иногда они встречаются и требуют адекватного представления в схеме БД.

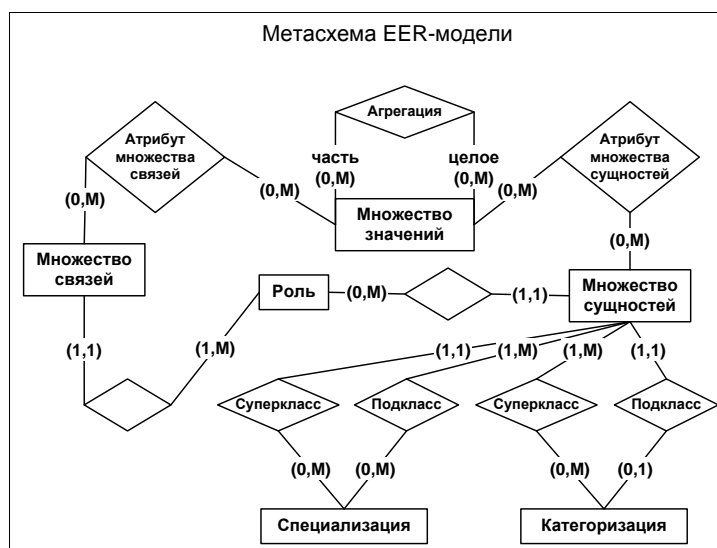
Начнем пояснение понятия «категоризация» с констатации отличия характера отношений между подклассами и суперклассами специализаций и категоризаций. Если для первых это отношение характеризуется словами «есть некоторый», и отображение из подкласса в суперкласс обязательно полное функциональное, то для категоризаций это отношение соответствует фразе «есть некоторый или А, или Б, или ...», и в силу этого указанное отображение частичное функциональное.

Вторым отличием этих понятий является то, что все классы специализации представляют собой «родственные» классы, имеющие в содержании соответствующих им понятий общие условия. Более того, определения понятий подклассов ссылаются на родительское понятие: «Множество – это комплекс, в котором ...», «Кортеж – это комплекс, в котором ...». Суперклассы категоризации, как правило, абсолютно различные, всегда непересекающиеся классы. Действительно, что общего между гражданином и организацией, или между гражданином и территориальным образованием типа город или край. Категория *СУБЪЕКТ ПРАВА* является всего лишь механистическим объединением некоторых представителей суперклассов, и это явно указывается в ее определении – «Субъект права – это либо физическое лицо (гражданин), либо юридическое лицо (организация), либо административно-территориальное образование, такое, что ...».

Кто-то может заметить, что если перевернуть категоризацию, то получится специализация. Давайте сделаем это с нашей категоризацией.

Обе конструкции представлены на слайде. Проверим, справедливы ли свойства специализации для вновь полученной структуры. Отношение «есть некоторый», по крайней мере, для класса граждан выполняется отнюдь не всегда. Отображение из множества граждан во множество субъектов права не является полным функциональным. Определение гражданина, как субъекта права, обладающего какими-то особыми свойствами, тоже не годится.

Несмотря на кажущееся сходство, налицо явно не специализация. Для таких случаев и предусмотрена категоризация.



Как вы уже догадались семантические модели данных с бедным понятийным базисом, удаляющие нас от идеальной семантической модели данных, не представляют для нас особого интереса. Их метасхемы будут лишь подсхемами метасхемы ER-модели Чена. А вот EER-модель заслуживает того, чтобы ее метасхема была здесь приведена.

Сразу бросается в глаза отличие от метасхемы ER-модели – в этой метасхеме появились множества сущностей *СПЕЦИАЛИЗАЦИЯ* и *КАТЕГОРИЗАЦИЯ*, а также соответствующие множества связей *СУПЕРКЛАСС* и *ПОДКЛАСС* между ними и множеством сущностей *МНОЖЕСТВО СУЩНОСТЕЙ*. Кстати, обратите внимание на отличие в кардинальных числах вновь образовавшихся отображений. Конкретное множество сущностей может неоднократно использоваться в разных специализациях, причем как в роли суперкласса, так и в роли подкласса. А вот категорией (подклассом категоризации) оно может быть лишь однажды. Вот вам еще одно отличие между специализациями и категоризациями.

А в остальном между метасхемами полное сходство, что еще раз убеждает в том, что концепции ER-модели Чена остались в EER-модели неизблемыми.



К сожалению ни сама ER-модель Чена, ни ее расширение в лице EER-модели не удостоились чести быть реализованными в CASE-инструментах. Разработчики этих систем почему-то предпочли им более бедные с точки зрения выразительных средств модели, которые, тем не менее, гордо именуются моделями «сущность-связь».

В этом плане в лучшую сторону отличается модель, реализованная в Oracle Designer и представляющая собой широко известную нотацию Баркера. По сравнению с EER-моделью в ней отсутствуют:

- множества связей степени больше двух,
- атрибуты множеств связей,
- многозначные атрибуты,
- атрибутные отображения в Декартово произведение множеств значений.

Давайте проанализируем, какие «технические решения» предстоит предпринимать проектировщику схемы БД, выбравшему в качестве семантической модели ER-модель Баркера.

Если ему необходимо представить множество связей степени n , придется использовать вспомогательное множество сущностей и n бинарных множеств связей. Одно из отображений каждого такого множества связей (для которого вновь созданное множество сущностей является ООО) будет полным функциональным. Типы обратных им отображений совпадают с типами отображений, определяемых ролями множества связей степени n .

В случае наличия атрибутов у множества связей универсальным решением является замена этого множества связей множеством сущностей и бинарными множествами связей. Однозначные атрибуты бывшего множества связей станут атрибутами нового множества сущностей. Другим решением может стать добавление атрибутов множества связей в то множество сущностей, участвующее в этом множестве связей, роль которого определяет функциональное отображение. Это решение становится идеальным, если это отображение к тому же полно.

Многозначные атрибуты множеств сущностей и множеств связей потребуют создания для каждого такого атрибута самостоятельного множества сущностей с атрибутом для значений и бинарного множества связей типа $M:1$ между новым множеством сущностей и множеством сущностей – владельцем многозначного атрибута. Очевидно, что в таком случае множество связей – владельца многозначного атрибута необходимо превратить во множество сущностей.

В случае наличия атрибутных отображений в Декартово произведение множеств значений их надо представлять или элементарными атрибутами, или атрибутами-агрегатами, или и тем, и другим одновременно.

Претерпела изменения и графическая нотация. Коль скоро исчезли связи степени больше двух и атрибуты связей, оставшиеся бинарные множества связей можно представлять обычным ребром с двумя концами, каждый из которых соответствует одной роли, о чем говорят пометки на концах. Различные графические примитивы, которыми изображаются концы ребер, говорят о минимальных и максимальных кардинальных числах отображений (их в бинарном множестве связей как раз два):

- пунктирная линия, выходящая из множества сущностей, означает, что в контексте данной связи эти сущности являются регулярными (МинКЧ = 0);
- сплошная линия говорит о слабости соответствующих сущностей в контексте данной связи (МинКЧ = 1);
- одиночный конец расположен у множества сущностей, являющегося областью значений функционального отображения (МаксКЧ = 1);
- вилочкообразный («воронья лапка») конец примыкает к множеству сущностей, являющемуся областью значений нефункционального отображения (МаксКЧ = ∞).

Определенную сложность для людей, «обремененных» умением анализировать отображения, может представлять то, что информация о количественных характеристиках одного отображения между множествами сущностей в нотации Баркера разбросана по разным концам ребра множества связей. Рассмотрим, например, множество связей *БОЛЬНИЧНАЯ ПАЛАТА*. Отображение *ПАЛАТА* → *БОЛЬНИЦА*, как мы знаем, является полным функциональным (МинКЧ = МаксКЧ = 1). Однако, информация о том, что МинКЧ = 1, представлена сплошной линией на конце ребра, расположенном возле множества сущностей *ПАЛАТА*, а о том, что МаксКЧ этого же отображения равен 1 – одиночным концом возле множества сущностей *БОЛЬНИЦА*.

Очевидно, Баркер при выборе выразительных средств для своей нотации не пользовался аппаратом отображений, а посчитал, что так будет лучше для восприятия этой информации человеком в психологическом плане.

Следующие знаки на концах ребер передают дополнительную информацию о связях:

- «|» (поперечный штрих) говорит об ID-зависимости (например, он стоит на ребре множества связей *БОЛЬНИЧНАЯ ПАЛАТА* возле множества сущностей *ПАЛАТА*);
- «◇» (ромбик) говорит о том, что указанный им конец связи неперемещаемый (например, при переходе врача из одной больницы в другую, можно изменить указатель на больницу, не удаляя связь; но нельзя менять указатель на врача в связях типа *ШТАТНЫЙ ВРАЧ*).

Сами множества сущностей представляются в ER-диаграммах прямоугольниками с закругленными углами, внутри которых указывается имя множества сущностей и имена его атрибутов вместе с некоторыми ограничениями целостности. Символы, указанные слева от имен атрибутов множества сущностей, означают следующее:

- «#» – атрибут участвует в уникальной идентификации сущностей;
- «*» – обязательный атрибут (*NOT NULL*);
- «○» – необязательный атрибут (*NULL*).

Своеобразно в нотации Баркера задаются на ER-диаграммах и специализации (категоризации отсутствуют). Во-первых, из всех возможных видов специализаций допустимы только полные и непересекающиеся. Во-вторых, графически они представляются как вложенные в прямоугольник родительского множества сущностей (суперкласс) прямоугольники дочерних множеств сущностей (подклассы). В-третьих, каждое множество сущностей может лишь однажды выступить как в роли суперкласса, так и в роли подкласса, хотя может одновременно совмещать обе эти роли.

На слайде представлена ER-диаграмма для медицинской ПрО в нотации Баркера.



И уж совсем ER-модель низведена до уровня реляционной модели в нотации IDEF1X, широко представленной во многих CASE-инструментах. В ней к уже упоминавшимся в нотации Баркера изъятиям добавились следующие требования:

- бинарные множества связей типа $M:N$ необходимо сводить вручную к дополнительному множеству сущностей и паре множеств связей типа $1:M$;
- необходимо дублировать первичные ключи множеств сущностей (и формировать, таким образом, внешние ключи) для представления множеств связей типа $1:M$.

Указанные требования фактически превращают ER-модель в нотации IDEF1X в реляционную модель. Для того, чтобы убедиться в этом, достаточно сравнить схемы БД, представленные на этом и предыдущем слайде.

Понятно, что реализация CASE-средства с такой моделью особых трудностей не вызывает. В нем не придется делать даже тех преобразований, которые осуществляются в Oracle Designer.

Вопросы и задания к параграфу 3.2

1. В чем отличие сущностей и связей ER-модели от предметов ПрО и отношений между ними?
2. Какого типа отношение существует между предметами и их сущностями?
3. Как соотносится определение атрибута, данное Ченом, с определением этого же термина во второй главе настоящего пособия?
4. Чем отличается представление связей в БД ER-модели и реляционной модели?
5. По каким правилам строится ER-диаграмма?
6. В каких различных формах может быть представлено в ER-модели некоторое явление ПрО?
7. Как осуществлять выбор подходящей формы?
8. Приведите свои примеры рекурсивных связей, связей степени большей 2.
9. Как вы думаете, почему на ER-диаграммах обычно не показывают атрибуты и множества значений?
10. Какие формы предложил Чен для представления БД в своей модели? Опишите их.
11. Что является данными, когда рассматривается метасхема?
12. В чем особенности синтаксиса ограничений целостности на значения атрибутов в ER-модели?
13. Поясните особенности понятия «ключ» в ER-модели.
14. Почему в ER-модели неуместно использование первичных ключей?
15. Объясните и приведите примеры всех способов идентификации связей.

16. Что представляют собой E-зависимость и ID-зависимость?
17. Чем отличаются множество слабых сущностей и множество регулярных сущностей?
18. Поясните, как расставлять пометки на ребрах ролей ER-диаграммы в случае множества связей степени больше 2-х.
19. Укажите особенности навигационного манипулирования данными в ER-модели.
20. Какие формы может принимать подграф запроса на языке CABLE?
21. В каких случаях можно явно задавать в запросе теоретико-множественные операции?
22. Поясните критерии качества семантических моделей.
23. Нарисуйте общую схему модификаций ER-модели Чена. Какие концепции добавлялись или удалялись на каждом из этапов этих модификаций?
24. Чем чревато использование невыразительных семантических моделей?
25. В каких случаях используются специализации и категоризации?
26. Какие ограничения целостности касаются специализаций и категоризаций?
27. Опишите графическую нотацию, применяемую для представления специализаций и категоризаций на EER-диаграммах.
28. Докажите, что понятия «специализация» и «категоризация» не тождественны.
29. Какие изменения произошли в метасхеме EER-модели по сравнению с метасхемой ER-модели?
30. Каких структурных понятий мы лишаемся при переходе от EER-модели к ER-модели Баркера?
31. Какими структурными понятиями ER-модели Баркера мы вынуждены их заменять?
32. Опишите графическую нотацию Баркера.
33. Укажите правила трансформации схемы БД из ER-модели Баркера в реляционную модель.
34. Охарактеризуйте ER-модель в нотации IDEF1X.