

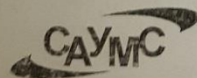
ФГБОУ ВО «Воронежский государственный
технический университет»

Кафедра системного анализа
и управления в медицинских системах

500-2015

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторной работы № 1
по дисциплине «Базы и банки данных медицинских систем»
для студентов направления
12.03.04 «Биотехнические системы и технологии»
(профиль «Биотехнические и медицинские аппараты и системы»)
очной формы обучения



Воронеж 2015

Составитель канд. техн. наук, доцент Е.И. Новикова

УДК 681.3

Методические указания к выполнению лабораторных работ № 1 по дисциплине «Базы и банки данных медицинских систем» для студентов направления 12.03.04 «Биотехнические системы и технологии» (профиль «Биотехнические и медицинские аппараты и системы») очной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. Е.И. Новикова. Воронеж, 2015. 20 с.

Данное методическое указание предназначено для проведения лабораторной работы по разработке моделей данных «сущность-связь».

Предназначены для студентов 3 курса.

Ил. 14. Библиогр.: 2 назв.

Рецензент д-р техн. наук, проф. Е.Н. Коровин

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. О.В. Родионов

Печатается по решению редакционно-издательского совета Воронежского государственного технического университета

© ФГБОУ ВО «Воронежский
государственный технический
университет», 2015

РАЗРАБОТКА МОДЕЛЕЙ «СУЩНОСТЬ-СВЯЗЬ» НА ОСНОВЕ СТАНДАРТА IDEF1X

1. ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

Целью лабораторной работы является приобретение навыков по разработке моделей данных «сущность-связь».

2. СОСТАВ ИСПОЛЬЗУЕМОГО ОБОРУДОВАНИЯ

Используемые программно-аппаратные средства: персональное ЭВМ класса IBMPC стандартной конфигурации, программное средство для моделирования «сущность-связь» AllFusion ERWin Data Modelier by CA, MicrosoftOffice.

3. ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

3.1. Модель «сущность-связь»

Моделирование «сущность-связь» (ER-моделирование) – тип логического моделирования данных, который исходит из представлений, что все объекты информационной системы могут быть

обобщены в основные образы – абстрактные понятия, *сущности*, которые могут быть описаны через их характеристики (*атрибуты*). Сущности связаны между собой через действия, которые они выполняют по отношению друг к другу. Эти действия устанавливают отношения (*связи*) между сущностями.

Модель «сущность-связь» (entity-relationship model, ER-модель) представляет собой совокупность понятий и графических символов для построения концептуальных схем. Изначально модель ER была разработана П. Ченом в 1976 году с целью упрощения задачи проектирования баз данных. В 1993 году была создана версия модели

«сущность-связь» под названием IDEF1X («Integrated Definition 1 Extended») – «Единый стандарт, версия 1, расширенная»), которая была объявлена национальным стандартом США, и в настоящее время является наиболее часто используемой, и большинство программных продуктов для моделирования данных основаны именно на этом стандарте.

В настоящее время существует ряд CASE-инструментов, предназначенных для разработки ER-моделей, таких как, например, AllFusion ERWin Data Modelier.

Сущность – это некоторый объект или процесс, идентифицируемый в рабочей среде пользователя, данные о котором пользователи хотели бы отслеживать (хранить) в базе данных. Например, сотрудник Иванов И.И., продукт с кодом 8593539097211, заказ 10248 и т.п. Однотипные сущности объединяются в классы сущностей. Например, все сотрудники объединяются в один класс сущностей СОТРУДНИКИ. Класс сущностей называют чаще всего просто сущностью. Сущности в IDEF1X обозначаются в виде прямоугольников с прямыми или закругленными углами.

Экземпляр сущности представляет собой отдельный элемент класса сущностей.

Сущности имеют некоторый набор *атрибутов*, или свойств, которые описывают характеристики сущности. Например, сущность СОТРУДНИКИ может быть охарактеризована таким набором атрибутов: Фамилия, Имя, Дата рождения, Адрес, Телефон, Дата найма и др. На диаграмме сущности могут изображаться без атрибутов (рис. 1 а), с указанием только тех атрибутов, которые составляют первичный ключ (рис. 1 б), или с указанием всех имеющихся атрибутов (рис. 1 в).

Чтобы настроить уровень отображения в AllFusion ERWin Data Modelier, щелкните правой кнопкой в свободном месте файла модели, в появившемся контекстном меню (рис. 2) выберите из пункта

Display level один из предлагаемых вариантов.

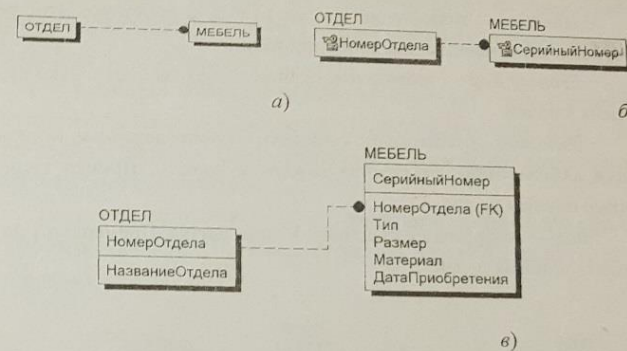


Рис. 1. Уровни детализации ER-диаграммы:
а) – только сущности; б) – сущности и первичные ключи;
в) – сущности и их атрибуты

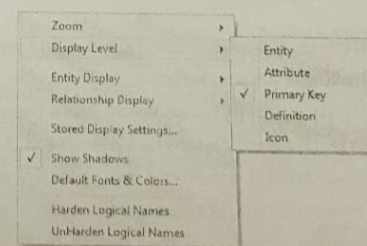


Рис. 2. Выбор уровня детализации модели

Особенности различных режимов отображения модели:

– Entity – режим отображения сущности, служит для удобства

обзора больших диаграмм или размещения прямоугольников сущностей на диаграмме на начальном этапе разработки модели.

– *Attribute* – отображение сущности с ее атрибутами. Этот режим является основным при проектировании.

– *Primary Key* – отображение сущности с указанием только первичных ключей.

– *Definition* – отображение сущности с ее определением, используется для описания назначения каждой сущности. Пример такого режима показан на рис. 3.

– *Icon* – отображение сущности в виде значка. Применяется для презентационных целей.

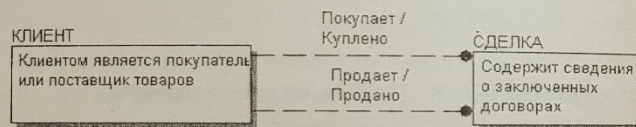


Рис. 3. Режим определения сущности

Атрибуты делятся на простые и составные, однозначные и многозначные, идентификаторы и дескрипторы.

Простые (атомарные) атрибуты не могут быть разделены на более мелкие компоненты. Пример: атрибуты Отчество и Пол.

Составные (композитные) атрибуты могут быть разделены на несколько простых компонентов, характеризующихся независимым существованием. Например, атрибут Адрес может быть разбит на отдельные атрибуты Улица, Район, Город, Почтовый индекс. Решение о моделировании составных атрибутов (разбивать на отдельные атрибуты или рассматривать как единое целое) зависит от проектировщика и от того, как впоследствии эти данные будут использоваться. Например, если в дальнейшем не предусматривается поиск по городу

или району, то атрибут Адрес можно рассматривать как единое целое. В случае, если часто требуется производить выборку по городу, району (или другим составным адреса), то атрибут Адрес целесообразнее рассматривать как составной.

Однозначный атрибут имеет одно значение для каждого экземпляра сущности. Например, каждый сотрудник имеет только одно имя, один номер паспорта и др. Таким образом, первичный ключ всегда является однозначным атрибутом.

Многозначный атрибут содержит несколько значений для одного экземпляра сущности. Например, сотрудник может иметь несколько телефонных номеров.

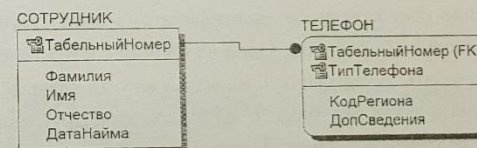


Рис. 4. Представление многозначных атрибутов

Многозначные атрибуты представляются в модели «сущность–связь» путем создания новой *идентификационно-зависимой* сущности и построения связи с ней «один ко многим». На рис. 4 показан пример реализации многозначного атрибута.

Если сущность отображается на диаграмме с полным набором атрибутов, то символизирующий ее прямоугольник делится на две части. В верхней части указываются атрибуты, входящие в *идентификатор сущности* (в реляционной модели – первичный ключ), а в нижней – все прочие атрибуты (*дескрипторы*).

Под *идентификатором* подразумевается элемент данных, который позволяет отличать один экземпляр сущности от другого. Выбор

идентификатора сущности должен осуществляться исходя из соотношений длины значений атрибутов, минимального количества атрибутов в ключе, а также наличия гарантий уникальности его значений в текущий момент времени и в обозримом будущем.

Идентификаторы сущностей бывают двух типов: *естественные* и *искусственные*. *Естественный идентификатор* сущности – это атрибут (или набор атрибутов), который уже присутствует в сущности и который уникально определяет каждый экземпляр этой сущности. Например, ИНН (индивидуальный налоговый номер) может быть естественным идентификатором сущности СОТРУДНИКИ. *Искусственный идентификатор* – это атрибут, добавляемый в сущность специально, чтобы она имела уникальный идентификатор.

Добавление искусственного идентификатора может быть необходимо по нескольким причинам. Одна из причин состоит в том, что может существовать естественный идентификатор, который будет слишком большим для практического использования. Другая причина добавления искусственного идентификатора в том, что сущность может и не иметь естественного идентификатора. Например, сущность КЛИЕНТЫ без атрибута КодКлиента не будет обладать атрибутом или группой атрибутов, уникально идентифицирующих каждый экземпляр сущности. Часто использование подобных ключей экономит время и гарантирует, что все экземпляры сущности будут уникальны. Такие ключи в базе данных часто имеют тип данных AUTOINCREMENT.

Идентификатор может быть *простым* или *составным*. Составной идентификатор включает в себя два или более атрибутов, а простой – один атрибут.

Добавление сущностей в файл модели осуществляется с помощью панели объектов (рис. 5). Как видно из рисунка, кроме сущностей из панели объектов доступны также и другие элементы ER-

модели. Если эта панель не видна, то ее можно отобразить, открыв меню View и выбрав пункт Toolbars, а затем из раскрывающегося списка – Toolbox.

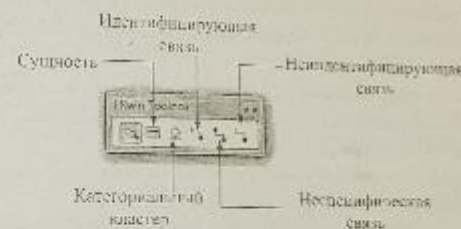


Рис. 5. Панель объектов IDENTITY

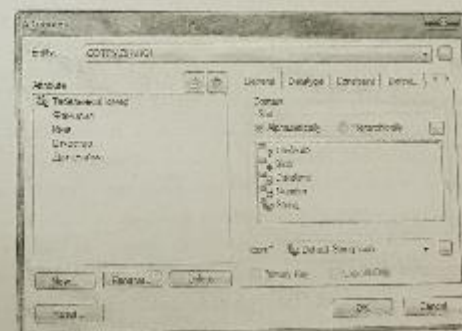


Рис. 6. Окно редактора атрибутов



Щелкнув правой кнопкой мыши по сущности, можно пойти в один из редакторов:

– редактор *Entity Properties* (в физической модели – *Table Properties*), связанный с сущностью в целом; здесь можно задать имя сущности (*Name*) и другие свойства;

– редактор атрибутов *Attributes (Columns)* – в физической модели), показанный на рис. 6, предназначен для добавления, удаления и редактирования атрибутов.

Чтобы добавить новый атрибут, нужно щелкнуть по кнопке *New*, для удаления предназначена команда *Delete*, а для изменения имени атрибута – *Rename* (рис. 6).

Атрибуты первичного ключа отмечаются флажком *Primary key*, расположенным в правой части редактора атрибутов.

Для изменения порядка следования атрибутов предназначены кнопки  – переместить выделенный атрибут вверх и  – переместить выделенный атрибут вниз.

Взаимоотношения сущностей выражаются *связями*. Связи в реляционной модели создаются путем помещения первичного ключа одной таблицы в другую таблицу. По отношению ко второй таблице такой ключ называется *внешним* (обозначается FK – Foreign Key).

Связи характеризуются *кардинальностью*, которая показывает количество возможных связей для каждого экземпляра сущностей участниц данной связи. Имеются три основных типа связей: «один к одному», «один ко многим» и «многие ко многим».

В связи «один к одному» (обозначается также 1:1) один экземпляр одной сущности связан с одним экземпляром другой сущности.

Связь «один ко многим» (другой вариант обозначения 1:N) означает, что один экземпляр одной сущности связан с множеством экземпляров другой сущности.

В связи «многие ко многим» (или M:N) множество экземпляров одной сущности может быть связано с множеством экземпляров другой сущности.

В стандарте IDEF1X определены четыре типа связей: неидентифицирующие, идентифицирующие, неспецифические и категориальные связи.

Неидентифицирующие связи – это связи принадлежности «один к одному» или «один ко многим» между двумя идентификационно-независимыми сущностями (отсюда характеристика «неидентифицирующие»).

Примечание. Связи принадлежности – это связи типа «имеет» (HAS-A relationship). Например, клиент *имеет* заказ, сотрудник *имеет* компьютер и т.д.

На рис. 1 был приведен пример неидентифицирующей связи. Согласно стандарту, неидентифицирующие связи изображаются пунктирной линией и рисуются в направлении от родительской сущности к дочерней. В случае связи «один ко многим» родителем является сущность на той стороне связи, которой соответствует «один». В примере, показанном на рис. 1, родительской является сущность ОТДЕЛ. В случае связи «один к одному» любая из сущностей может считаться родителем, но стандарт IDEF1X предписывает выбрать на эту роль какую-то одну из сущностей. Как видно из рис. 1, на конце линии связи рядом с сущностью-потомком (МЕБЕЛЬ) на диаграмме «сущность—связь» стандарта IDEF1X помещается закрашенный кружок.

По умолчанию неидентифицирующие связи принадлежности имеют кардинальность «один ко многим» с обязательным родителем и необязательным потомком. Такие связи изображаются пунктирной линией с закрашенным кружком на стороне потомка без каких-либо других обозначений. Так, связь между сущностями ОТДЕЛ и МЕБЕЛЬ на рис. 1 имеет вид «один ко многим»: ни один из отделов не обязан иметь мебель, но любая мебель, если таковая имеется, должна обязательно числиться за каким-либо отделом.

Если кардинальность связи отличается от предполагаемой по умолчанию, используются дополнительные обозначения. Если поток является обязательным, то рядом с кружком на соответствующем конце линии связи помещается символ Р (от positive — положительный; имеется в виду положительное число). Если родитель является необязательным, то на родительском конце линии связи рисуется ромбик. На рис. 7 связь между сущностями ОТДЕЛ и СОТРУДНИК имеет вид «один ко многим», причем любой отдел должен иметь, по крайней мере, одного сотрудника, но отдельно взятый сотрудник не обязан числиться в каком-либо отделе.

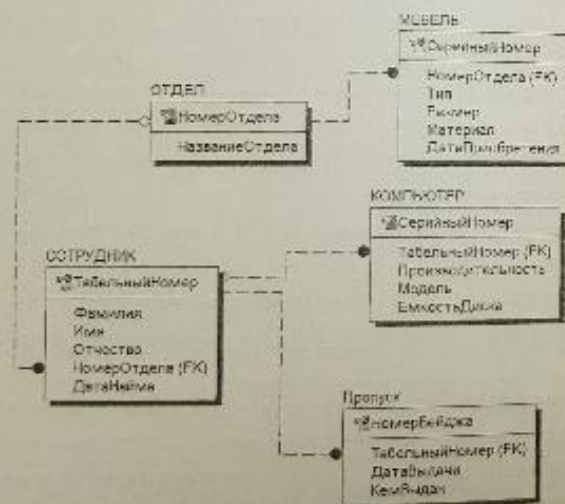


Рис. 7. Неидентифицирующие связи принадлежности

Связи «один к одному» обозначаются при помощи индексов рядом с кружком на соответствующей стороне связи. Число 1 показы-

васт, что требуется ровно один потомок, не больше и не меньше, а буква Z означает, что потомок может быть один, либо его может не быть вовсе.

Задать подобные параметры связи можно в окне *Relationships*. Чтобы его открыть, щелкните по линии связи правой кнопкой мыши и из контекстного меню выберите пункт *Relationship properties*, после этого откроется диалоговое окно, показанное на рис. 8.



Рис. 8. Определение параметров связи

В нижней части окна *Relationships* расположены две группы параметров связи: *Cardinality* (Кардинальность) и *Relationship Type* (Тип связи). По умолчанию кардинальность связи установлена на «Zero, one or more», т.е. один экземпляр родительской сущности может быть связан с числом экземпляров дочерней сущности от ну-

ля и более. Кардинальность связи можно изменить, отмечая соответствующий переключатель.

В группе *Relationship Type* можно изменить тип связи (идентифицирующая или неидентифицирующая), а также указать, может ли внешний ключ принимать значения Null. Если отмечена опция «No Nulls», то внешний ключ не может принимать значения Null, если отмечено «Nulls Allowed», то такие значения допускаются и на диаграмме эта опция подтверждается знаком ромбика на стороне родительской сущности.

На рис. 7 сущность СОТРУДНИК имеет связь с одной и только одной сущностью ПРОПУСК, а также с одной или нулевым количеством сущностей КОМПЬЮТЕР. Ромбик показывает, что компьютер не обязательно должен быть связан с каким-либо сотрудником (т.е. значение внешнего ключа ТабельныйНомер может принимать значения Null). Поскольку на линии связи ПРОПУСК-СОТРУДНИК со стороны сущности СОТРУДНИК ромба нет, каждый пропуск должен принадлежать какому-то сотруднику (в значениях внешнего ключа Null не допустимы).

Идентифицирующие связи – это связи принадлежности «один к одному» или «один ко многим» между сущностью-родителем и идентификационно-зависимой от нее дочерней сущностью, т.е. идентификатор родительской сущности является частью идентификатора дочерней сущности.

На рис. 9 сущность ЗДАНИЕ является родителем в идентификационно-зависимой связи с сущностью ОФИС. Обратите внимание, что идентификатор сущности ЗДАНИЕ, атрибут НомерЗдания, является частью идентификатора сущности ОФИС. Обозначение (FK) означает, что данный атрибут является внешним ключом другой сущности (в данном случае сущности ЗДАНИЕ). Идентифицирующие связи обозначаются сплошными линиями, а дочерние сущ-

ности в таких связях рисуются с закругленными углами. В данном примере никаких дополнительных обозначений рядом с закрашенным кружком нет, так что кардинальность связи предполагается заданной по умолчанию. Таким образом, в здании может располагаться ноль, один или множество офисов. Если бы рядом с кружком был указан индекс 1, то в здании мог бы располагаться ровно один офис, в случае индекса Z возможное количество офисов равнялось бы 0 или 1, а индекс P указывал бы на любое количество офисов, большее или равное 1. На родительской стороне идентифицирующей связи не может быть ромба, поскольку внешний ключ является частью первичного ключа, а, значит, для него недопустимы Null-значения.

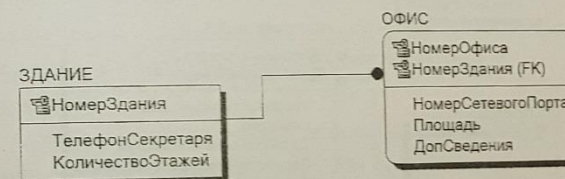


Рис. 9. Идентифицирующие связи принадлежности

Под *неспецифической связью* в IDEF1X понимается связь типа «многие ко многим». Неспецифические связи обозначаются закрашенными кружками на обоих концах сплошной линии связи (рис. 10).

В IDEF1X не предусмотрено задание минимальной кардинальности для неспецифической связи. Причиной является то, что такие связи не могут быть напрямую выражены в терминах реляционной модели (и, соответственно физически реализованы), и их применение только теоретическое. Связь «многие ко многим» на этапе логического моделирования нужно преобразовывать в две связи

атрибутов на соответствие требованиям к именованиям атрибутов деловой СУБД. В целом при составлении имен таблиц и столбцов желательно не использовать пробелы (лучше использовать знак подчеркивания или каждое слово в имени писать с заглавной буквы).

Одним из важнейших свойств столбца является обязательность или необязательность его значения. Это свойство должно быть определено в физической модели данных. Как видно из рис. 12, для каждого столбца, выделив его, можно задать требование NOT NULL. Если этот флажок отмечен, то предполагается, что ввод данных в этом столбце является обязательным (т.е. Null-значения не допустимы).



Рис. 12. Окно редактирования столбцов

4. ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ

1. Получить вариант задания у преподавателя, систему, для которой будете разрабатывать модель базы данных. Определите, какого типа информация должна храниться в этой базе данных. Определите перечень сущностей; атрибуты сущностей, связи между сущностями;

первичные и внешние ключи сущностей

2. Запустите программу AllFusion ER Win Data Modelier.
3. На главном экране выберите пункт меню File -> Open.

3. Далее откроется диалоговое окно создания нового файла модели, показанное на рис. 13. Здесь в разделе *New Model Type* отметить флажок *Logical/Physical*, чтобы обозначить возможность хранения физической и логической моделей в одном файле, а в разделе *Target Database* (Целевая СУБД) выберите Access и нажмите *OK*.

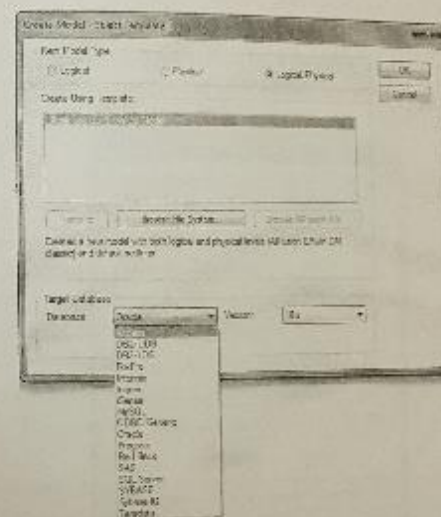


Рис. 13. Окно создания файла модели данных

4. После того, как откроется новый файл в режиме логической модели, перейдите к меню *Format*, выберите пункт *Default Fonts & Colors...*, чтобы открыть окно настройки шрифта и цвета модели (рис.

ских систем: учеб. пособие / Е.И. Новикова, О.В. Родионов. - Воронеж: ФГБОУ ВПО «Воронежский государственный технический университет», 2014. - 132 с.

2. Харрингтон, Дж. Л. Проектирование реляционных баз данных / Дж. Л. Харрингтон. - М.: Лори, 2006.

СОДЕРЖАНИЕ

РАЗРАБОТКА МОДЕЛЕЙ «СУЩНОСТЬ-СВЯЗЬ» НА ОСНОВЕ СТАНДАРТА IDEF1X

1. Цель лабораторной работы	1
2. Состав используемого оборудования	1
3. Теоретическое введение	1
4. Задание к лабораторной работе	16
5. Указания по оформлению отчета по лабораторной работе	19
6. Контрольные вопросы к лабораторной работе	19
Библиографический список	19

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторной работы № 1
по дисциплине «Базы и банки данных медицинских систем»
для студентов направления
12.03.04 «Биотехнические системы и технологии»,
(профиль «Биотехнические и медицинские аппараты и системы»
очной формы обучения

Составитель
Новикова Екатерина Ивановна

В авторской редакции

Подписано в печать 26. 11. 2015
Формат 60×84/16. Бумага для множительных аппаратов.
Усл. печ. л. 1,4. Уч.-изд. л. 1,2. Тираж 24 экз. «С» 82.
Зак. № 201.

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14