

запрос преобразуется в формальный запрос, понятный системе. Информационный язык ИПС может быть основан на подмножестве естественного языка, которое относится к обслуживаемой ПрО. Но чаще поиск документа осуществляется с помощью шаблонов – экранных форм, включающих поля описания документа. В эти поля вносятся конкретные значения, которые и определяют условия поиска документов. На рис. 1.2 приведён пример поиска через экранную форму в справочнике библиотек. В форме указаны два условия – округ и фрагмент названия библиотеки, и в результате поиска система выдала три записи, удовлетворяющие этим условиям.

Id	Номер	Название	Тип
259	43	им. Горького А.М.	
258	43	им. Горького А.М.	
248	23	им. Горького А.М.	Детская

Рис. 1.2. Пример поиска данных через экранную форму

Мы будем основное внимание уделять фактографическим АИС, имея в виду, что ИПС и документальные АИС создаются с помощью тех же программных средств и на тех же принципах, что и СОД, а специфические моменты обработки данных реализуются через приложения (программы, внешние по отношению к ядру СОД).

Разработка любой АИС начинается с определения предметной области.

1.3. Предметная область информационной системы

Предметная область (ПрО) информационной системы рассматривается как совокупность реальных процессов и объектов (**сущностей**), представляющих интерес для её пользователей [6]. Каждая из сущностей ПрО обладает определённым набором свойств (атрибутов).

Для упрощения процедуры описания ПрО в большинстве случаев прибегают к определению *типов сущностей*. Тип позволяет выделить из всего множества сущностей ПрО группу сущностей, однородных по структуре и поведению (относительно рамок рассматриваемой ПрО). Например, для ПрО "Институт" в качестве типов сущностей могут рассматриваться студенты, преподаватели, дисциплины и т.п.

Данные предметной области представляются *экземплярами сущностей* (студент Иванов, преподаватель Сидоров, дисциплина "Базы данных"). Экзем-

пляры сущностей одного типа обладают одинаковыми наборами атрибутов, но должны отличаться значением хотя бы одного атрибута для того, чтобы быть узнаваемыми (например, студенты могут иметь одинаковые ФИО, но должны иметь разные номера зачётных книжек).

Среди атрибутов сущности можно выделить существенные и малозначительные. Признание какого-либо свойства существенным носит относительный характер. Например, атрибут *Должность* для сотрудника является существенным, а для читателя библиотеки – малозначительным.

Атрибуты можно условно классифицировать следующим образом:

- 1) *Идентифицирующие и описательные атрибуты*. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа, описательные заключают в себе интересующие свойства сущности.
- 2) *Составные и простые атрибуты*. Простой атрибут состоит из одного компонента, его значение неделимо; составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных.
- 3) *Однозначные и многозначные атрибуты* (могут иметь соответственно одно или много значений для каждого экземпляра сущности).
- 4) *Основные и производные атрибуты*. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов.
- 5) *Обязательные и необязательные*. Значение обязательного атрибута всегда устанавливается при помещении данных в БД; значение необязательного атрибута может быть пропущено.

Спецификация атрибута состоит из его названия, типа данных, размера и описания ограничений целостности – множества значений, которые может принимать данный атрибут.

Между сущностями ПрО могут существовать **связи**, имеющие различный содержательный смысл (семантику). Например, студент *учится* в группе, врач *лечит* пациента, клиент *имеет* вклад в банке. Связи могут быть **факультативными** или **обязательными**. Если вновь порождённая сущность одного из типов оказывается по необходимости связанной с сущностью другого типа, то между этими типами сущностей есть обязательная связь. Иначе связь является факультативной. Примеры обязательной и факультативной связей приведены на рис. 1.3. Здесь связь *замещает* является обязательной (изображается двойной линией), потому что каждый сотрудник *должен* работать на определённой должности, а связь *замещается* является факультативной, т.к. должность *может быть* вакантна.

Примечание. В данном учебном пособии наименования сущностей, атрибутов и связей выделяются курсивом и подчёркиванием. Кроме того:

1. Сущность записывается прописными буквами (*ОТДЕЛ*).
2. Атрибут сущности начинается с прописной буквы (*Название*). Ключевой атрибут выделяется полужирным шрифтом (*Табельный номер*).
3. Связь между сущностями определяется глаголом (*работает*).

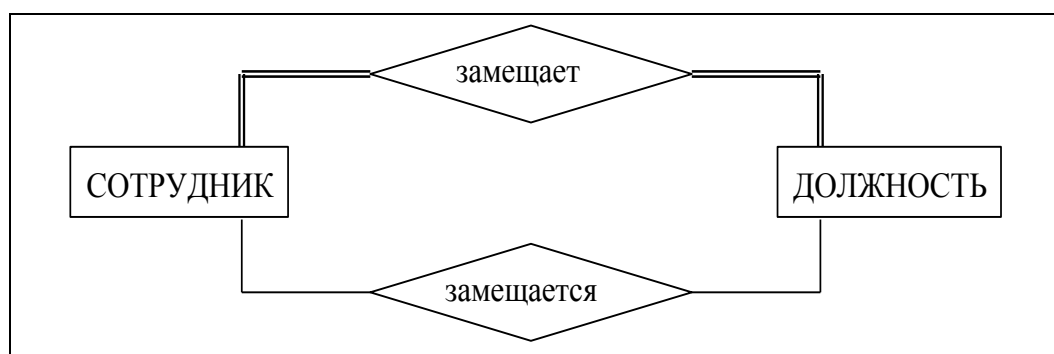


Рис. 1.3. Примеры обязательной и факультативной связей

Для удобства каждую связь между сущностями можно изображать одним ромбом (рис. 1.4). Выделяют также показатель **кардинальности связи**: "один к одному" (1:1), "один ко многим" (1:n) и "многие ко многим" (m:n) (рис. 1.4).

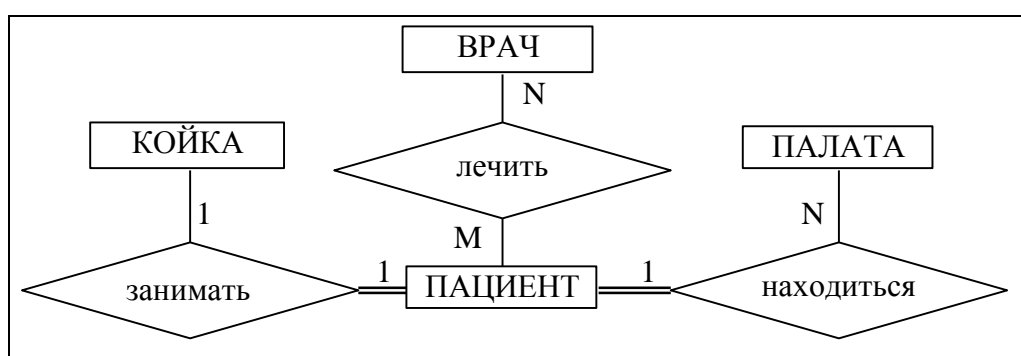


Рис. 1.4. Примеры различной кардинальности связей

Связи, приведённые на рис. 1.4, с учётом семантики означают следующее:

- пациент–койка (1:1) – каждый пациент занимает одну койку, каждая койка в каждый момент времени может быть занята только одним пациентом;
- палата–пациент (1:n) – каждый пациент находится в одной палате, в каждой палате могут находиться несколько пациентов;
- пациент–врач (n:m) – каждый пациент может лечиться у нескольких врачей, каждый врач может лечить несколько пациентов.

Обратите внимание: необязательная связь имеет модификатор "может", а у обязательной связи его нет.

Степень связи – это количество типов сущностей, которые входят в связь. Различают унарные (рис. 1.5,а), бинарные (рис. 1.5,б) и тернарные (рис. 1.5,в) связи. (На практике связи с большей степенью редко используются). Унарная связь означает, что одни экземпляры сущности связаны с другими экземплярами этой же сущности (например, одни сотрудники руководят другими, а деталь может являться частью механизма).

Различают *тип связи* и *экземпляр связи*. Тип связи определяется её именем, обязательностью, степенью и кардинальностью, например, бинарная связь *учится* между сущностями *ГРУППА* и *СТУДЕНТ*, обязательная для студента, кардинальностью 1:n. А экземпляр связи – это конкретная связь между студентом Сидоровым и группой Н-11, в которой он учится.

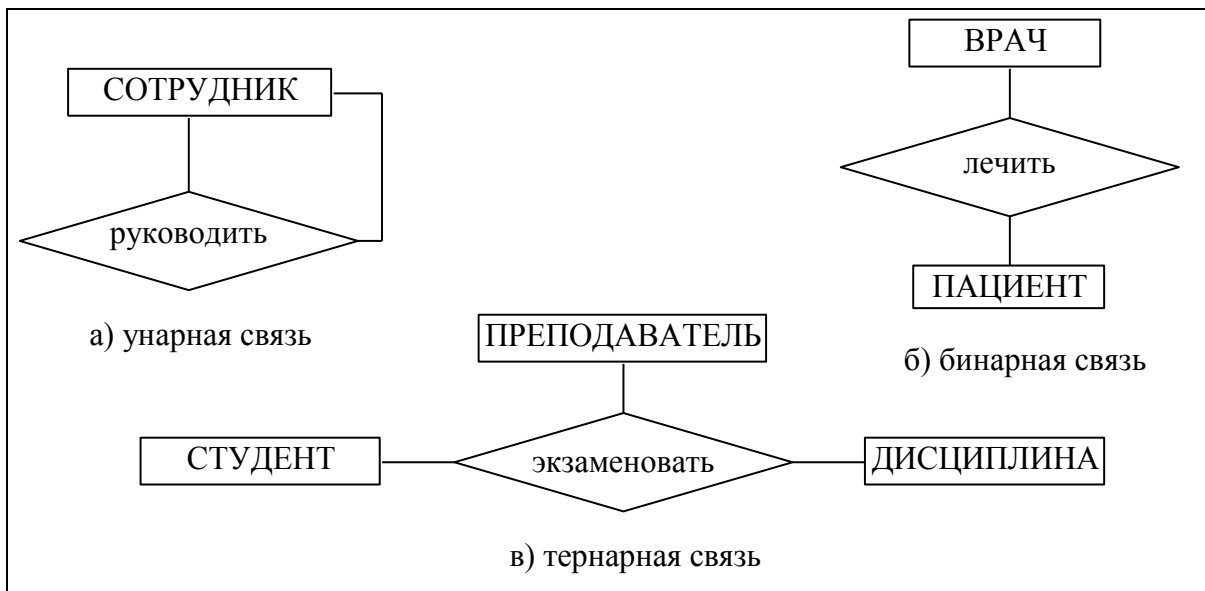


Рис. 1.5. Примеры связей различной степени

Совокупность типов сущностей и типов связей между ними характеризует структуру предметной области. Собственно данные представлены экземплярами сущностей и связей между ними. Данные экземпляров сущностей и связей хранятся в базе данных информационной системы, а описание типов сущностей и связей является метаданными.

Множества экземпляров сущностей, значения атрибутов сущностей и экземпляры связей между ними могут изменяться во времени. Поэтому каждому моменту времени можно сопоставить некоторое **состояние предметной области**. Состояния ПрО должны подчиняться совокупности правил, которые характеризуют семантику предметной области. В базе данных эти правила могут быть заданы с помощью так называемых *ограничений целостности*, которые накладываются на атрибуты сущностей, типы сущностей, типы связей и/или их экземпляры. Фактически ограничения целостности – это правила, которым должны удовлетворять значения данных в БД. Например, для библиотеки можно привести такие ограничения целостности: количество экземпляров книги не может быть отрицательным; номер паспорта читателя должен быть уникальным; каждая книга относится к определённому разделу рубрикатора ББК – библиотечно-библиографической классификации и т.д.

Для того чтобы обеспечить соответствие базы данных текущему состоянию предметной области, база данных *динамически обновляется* (периодически или в режиме реального времени). Это обновление называется **актуализацией данных**. Актуализация может проводиться:

- вручную, если изменения в данные вносит пользователь (например, запись сведений о выдаче абоненту книги в библиотеке);
- автоматизировано, если изменения инициируются пользователем, но выполняются программно (например, обновление списка должников в библиотеке – читателей, которые просрочили дату возврата книг);

- автоматически, если данные поступают в электронном виде и обрабатываются программой без участия человека (это касается, например, автоматизированных систем управления производством).

Правильность обновлений может контролироваться программно, но правильнее контролировать их автоматически с помощью ограничений целостности БД.

База данных является информационной моделью внешнего мира, некоторой предметной области. Во внешнем мире сущности ПрО взаимосвязаны, поэтому в БД эти связи должны быть отражены. Если связи между данными в БД отсутствуют, то имеет смысл говорить о нескольких независимых БД и хранить их отдельно.

1.4. Назначение и основные компоненты системы баз данных

Система БД включает два основных компонента: собственно базу данных и систему управления базами данных – СУБД (рис. 1.6). Большинство СОД включают также программы обработки данных (прикладное программное обеспечение, ППО), которые обращаются к данным через СУБД.

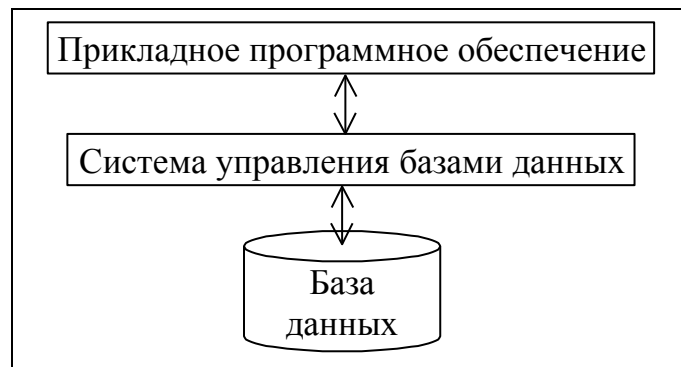


Рис. 1.6. Компоненты системы баз данных

В соответствии с рис. 1.6 СУБД обеспечивает выполнение двух групп функций:

- предоставление доступа к базе данных прикладному программному обеспечению (или квалифицированным пользователям);
- управление хранением и обработкой данных в БД.

Таким образом, обращение к базе данных возможно только через СУБД.

БД предназначена для хранения данных информационной системы. Пользователи обращаются к базе данных обычно не напрямую через средства СУБД, а с помощью внешнего интерфейса – приложения, входящего в состав АИС. Если пользователей можно разделить на группы по характеру решаемых задач, то приложений может быть несколько (по количеству задач или групп пользователей). Например, для библиотеки можно выделить три группы пользователей: читатели, которым нужно осуществлять поиск книг по различным признакам; сотрудники, выдающие и принимающие у читателей книги (библиотекари); сотрудники отдела комплектации, осуществляющие приём новых книг и списание старых.

1.5. Уровни представления данных

Современная технология баз данных основана на концепции многоуровневой архитектуры СУБД. Эти идеи впервые были сформулированы в отчёте рабочей группы по базам данных Комитета по планированию стандартов Американского национального института стандартов (ANSI/X3/SPARC). Этот отчёт был опубликован в 1975 г. В нём была предложена обобщенная трёхуровневая модель архитектуры СУБД, включающая концептуальный, внешний и внутренний уровни (рис. 1.7).



Рис. 1.7. Уровни представления данных

Концептуальный уровень архитектуры ANSI/SPARC служит для поддержки единого взгляда на базу данных, общего для всех её приложений и независимого от них и от среды хранения [5]. Концептуальный уровень представляет собой формализованную информационно-логическую модель ПрО. Описание этого представления называется *концептуальной схемой* или *схемой БД*.

Схема базы данных – это описание базы данных в терминах конкретной модели данных.

Внутренний уровень архитектуры поддерживает представление данных в среде хранения и пути доступа к ним [5]. На этом архитектурном уровне БД представлена в полностью "материализованном" виде, тогда как на других уровнях идёт работа на уровне отдельных экземпляров или множества экземпляров данных. Описание БД на внутреннем уровне называется *внутренней схемой* или *схемой хранения*.

Внешний уровень архитектуры БД предназначен для групп пользователей. Описание представления данных для группы пользователей называется *внешней схемой*. Наличие внешнего уровня позволяет поддерживать разное представление одних и тех же данных для различных групп пользователей или задач [5].

Каждый из этих уровней может считаться управляемым, если он обладает внешним интерфейсом, который обеспечивает возможности определения данных. В этом случае становятся возможными формирование и системная поддержка независимого взгляда на БД для какой-либо группы персонала или пользователей, взаимодействующих с БД через интерфейс данного уровня.

В архитектурной модели ANSI/SPARC предполагается наличие в СУБД механизмов, обеспечивающих междууровневое отображение данных "внешний – концептуальный" и "концептуальный – внутренний". Функциональные воз-

возможности этих механизмов определяют степень независимости данных на всех уровнях. На переходе "внешний – концептуальный" обеспечивается **логическая независимость данных**, на переходе "концептуальный – внутренний" – физическая независимость. Под логической независимостью подразумевается возможность вносить изменения в концептуальный уровень, не меняя представление БД для пользователей, или изменять представление данных для пользователей без изменения концептуальной схемы. **Физическая независимость данных** подразумевает возможность вносить изменения в схему хранения, не меняя концептуальную схему БД.

Основной характеристикой баз данных является совместное использование данных многими пользователями АИС. Должно существовать какое-то общее понимание информации, представленной данными. Общее понимание должно относиться к чему-либо внешнему по отношению к пользователям, и оно должно быть зафиксировано. Для этого необходима некоторая предварительно определённая грамматика, которую принято называть моделью данных.

! Задания для практических занятий

• **Тема: предметная область информационной системы**

1. Выделить базовые и зависимые сущности для различных ПрО ("Отдел кадров", "Магазин", "Институт", "Проектная организация").
2. Определить набор атрибутов для различных сущностей ("Студент", "Сотрудник", "Проект").
3. Определить связи между сущностями в различных ПрО ("Отдел кадров", "Магазин", "Институт", "Проектная организация").