

ИВАН АНДРЕЕВИЧ ТРЕЩЕВ  
ЕКАТЕРИНА СЕРГЕЕВНА КУДРЯШОВА

---

# Базы данных. Учебное пособие

ДЛЯ СТУДЕНТОВ

**Иван Андреевич Трещев  
Екатерина Сергеевна Кудряшова  
Базы данных. Учебное  
пособие. Для студентов**

*[http://www.litres.ru/pages/biblio\\_book/?art=41830596](http://www.litres.ru/pages/biblio_book/?art=41830596)  
ISBN 9785449645425*

**Аннотация**

В книге кратко рассмотрены теоретические вопросы проектирования баз данных. Приведены примеры лабораторных работ и расчетно-графического задания, опробированные в учебном процессе ВУЗа. Книга будет полезна как преподавателям, так и студентам, а также всем заинтересованным в проектировании, реализации и тестировании информационных систем.

# Содержание

Введение	5
Лекции	7
Лекция 1—3	7
Конец ознакомительного фрагмента.	28

# **Базы данных. Учебное пособие Для студентов**

**Иван Андреевич Трещев  
Екатерина Сергеевна  
Кудряшова**

*«Сложная система, спроектированная наспех,  
никогда не работает, и исправить её, чтобы  
заставить работать, невозможно». Закон Мерфи*

*Тестирование баз данных* Анастасия Сергеевна Ватолина

© Иван Андреевич Трещев, 2019

© Екатерина Сергеевна Кудряшова, 2019

ISBN 978-5-4496-4542-5

Создано в интеллектуальной издательской системе Ridero

# Введение

Любая информационная система сегодня в обязательном порядке включает в себя базу данных. Это может быть и база данных персонала и база данных контрагентов, товаров, услуг. Хотя большинство предприятий используют электронные таблицы, а не системы управления базами данных, но в ближайшем будущем все однозначно изменится. Использование электронных таблиц имеет свои неоспоримые преимущества в случае, если необходимо организовать небольшой массив данных и работать с ним приходится не часто. В случае же если «чистый» размер данных превосходит 100 Мб, то обработка таких массивов даже на современных ЭВМ с использованием электронных таблиц будет через мерно долгой.

СУБД эволюционируют вместе с данными. Сегодня OLAP кубы уже не являются ноу хау, а скорее превратились в обыденность, хотя еще 5 лет назад можно было по пальцам сосчитать предприятия использующие Oracle.

Отметим и все возрастающую роль распределенной обработки данных, поскольку когда возникают проблемы связанные с big data, даже хранение массивов в несколько десятков терабайт не представляется возможным на одной ЭВМ (исключая конечно специализированные системы хранения данных).

В данном пособии автор заостряет внимание на использовании СУБД Microsoft SQL Server и MySQL, поскольку первая используется во многих компаниях как стандарт де-факто, а вторая является свободно распространяемой и входит в комплект практически любого дистрибутива Unix-подобных операционных систем, дополнительно прекрасно интегрируется с PHP позволяя создавать платформы и приложения не только desktop но и web.

Все товарные знаки указанные в книге являются собственностью их правообладателей, а совпадения имен, названий, наименований и прочего – чистой случайностью.

# Лекции

## Лекция 1—3

**База данных** – это совокупность взаимосвязанных данных, находящихся под управлением системы управления базой данных (СУБД).

**Система управления базой данных** – совокупность языковых и программных средств, облегчающих для пользователей выполнение всех операций, связанных с организацией хранения данных, их корректировки и доступа к ним.

**Этапы проектирования реляционной БД декомпозиционным методом**

1) Разрабатывается универсальное отношение для БД (в универсальное отношение включаются все атрибуты, представляющие интерес для данного проектирования).

2) Определяются все функциональные зависимости между атрибутами данного отношения.

3) Определяется, находится ли отношение в нормальной форме Бойса – Кодда. Если да, то проектирование завершается, если нет, то осуществляется декомпозиция, т.е. разбиение отношения.

4) Шаги 2) и 3) повторяются для каждого нового отношения, полученного в результате декомпозиции. Проекти-

рование завершается, когда все отношения будут находиться в НФБК.

**Ключ** – поле (атрибут), по которому можно однозначно определить каждую запись в таблице.

**Функциональная зависимость** –  $(A \rightarrow B)$  атрибут  $B$  функционально зависит от атрибута  $A$  если в любой момент времени каждому значению атрибута  $A$  соответствует одно значение атрибута  $B$ .

Атрибут  $B$  **функционально полно** зависит от атрибута  $A$  если  $B$  не зависит ни от какого подмножества  $A$ .

**Нормализация** – процесс построения оптимальной структуры таблиц и связи между ними.

Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, модификации и удалении данных, чем все остальные наборы таблиц, с помощью которых могут быть представлены те же данные.

**Нормализованным отношением** называют отношение, каждое поле которого содержит только атомарные значения. (пример про ФИО)

Определение **первой нормальной формы** (1НФ): отношение  $r$  находится в 1НФ, если каждый его элемент имеет и всегда будет иметь атомарное значение. (Это определение просто устанавливает тот факт, что любое нормализованное отношение находится в 1НФ.)

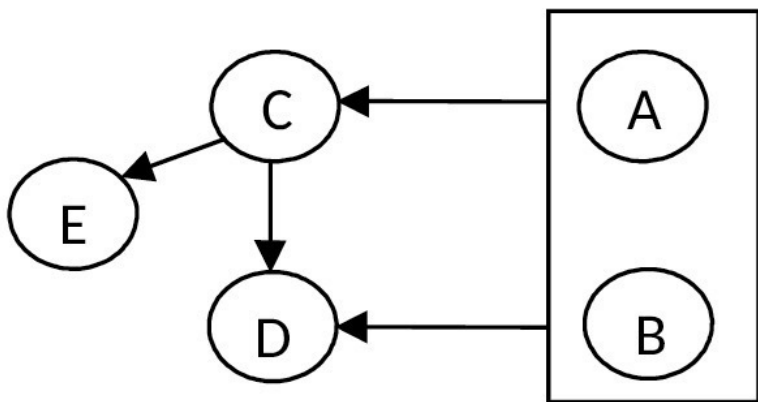
Определение **второй нормальной формы** (2НФ): отно-



шение  $r$  находится во 2НФ, если оно находится в 1НФ и если каждый его атрибут, не являющийся основным атрибутом, функционально полно зависит от первичного ключа этого отношения.

Определение **третьей нормальной формы (3НФ)**: отношение  $r$  находится в 3НФ, если оно является отношением во 2НФ и каждый его атрибут, не являющийся основным, не транзитивно зависит от первичного ключа этого отношения.

Транзитивная зависимость определяется следующим образом: если  $X \rightarrow Y$  и  $Y \rightarrow Z$ , то  $X \rightarrow Z$  ( $Z$  транзитивно зависит от  $X$ ).



$A, B \rightarrow C$

$A, B \rightarrow D$

$C \rightarrow D$

$C \rightarrow E$

Первичный ключ:  $AB$ .

Отношение находится в **1НФ**, поскольку все атрибуты имеют атомарные значения.

Отношение находится во **2НФ**, т.к. все атрибуты функционально полно зависят от первичного ключа отношения.

Так как  $A, B \rightarrow C$ ;  $C \rightarrow E$ , т. е.  $E$  транзитивно зависит от первичного ключа, значит отношение не находится в **3НФ**.

Отношение находится в **НФБК** если каждый детерминант отношения является его возможным ключом.

Детерминант – это атрибут, от которого зависит другой атрибут.

Отношение  $r$  находится в **4НФ** тогда и только тогда, когда при существовании многозначной зависимости в  $r$  атрибута  $Y$  от атрибута  $X$ , все остальные атрибуты  $r$  функционально зависят от  $X$ .

Атрибут  $X$  **многозначно** определяет атрибут  $Y$ , если с каждым значением  $x$  может использоваться значение  $y$  из фиксированного подмножества значений  $Y$ . Обозначается:  $X \twoheadrightarrow Y$ .

**Избыточной** функциональной зависимостью называют зависимость, заключающую в себе такую информацию, которая может быть получена на основе других зависимостей из числа использованных при проектировании БД.

Пусть  $r$  – отношение со схемой  $R$ ,

$w, x, y, z$  – подмножества  $R$ .

1-я аксиома вывода. **Рефлексивность**.

В  $r$  всегда имеет место  $X \rightarrow X$

2-я аксиома вывода. **Пополнение**.

Если  $r$  удовлетворяет  $X \rightarrow Y$ , то  $r$  удовлетворяет  $F$ -зависимости  $XZ \rightarrow Y$

3-я аксиома вывода. **Аддитивность** (так же известна под названием – объединение).

Если отношение  $r$  удовлетворяет  $X \rightarrow Y$  и  $X \rightarrow Z$ , то  $r$  удовлетворяет  $F$ -зависимости  $X \rightarrow YZ$ . (можно объединить правые части)

4-я аксиома вывода. **Проективность**.

Если отношение  $r$  удовлетворяет  $X \rightarrow YZ$ , то  $r$  удовлетворяет  $X \rightarrow Y$  и  $X \rightarrow Z$ .

(разбиваем совокупность)

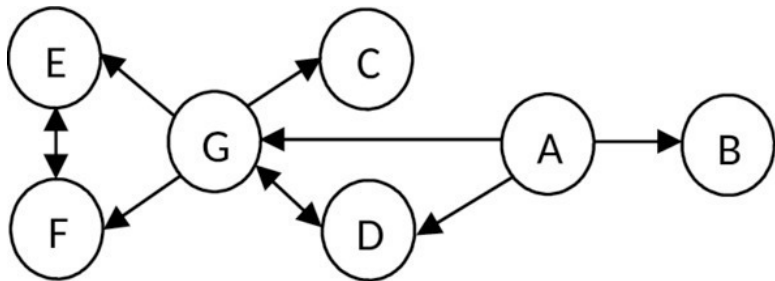
5-я аксиома вывода. **Транзитивность**.

$X \rightarrow Y$  и  $Y \rightarrow Z$  влечет за собой  $X \rightarrow Z$ . (избыточная транзитивная зависимость может быть удалена)

6-я аксиома вывода. **Псевдотранзитивность**.

Если  $r$  удовлетворяет зависимостям  $X \rightarrow Y$  и  $YZ \rightarrow W$ , то  $r$  удовлетворяет  $XZ \rightarrow W$ .

Исходная диаграмма функциональных зависимостей:



$A \rightarrow B$ ;  $A \rightarrow D$ ;  $A \rightarrow G$ ,  
 $D \rightarrow G$ ,  
 $G \rightarrow D$ ;  $G \rightarrow C$ ;  $G \rightarrow F$ ;  $G \rightarrow E$ ,  
 $E \rightarrow F$ ,  
 $F \rightarrow E$

Удалим из исходного набора функциональных зависимостей все избыточные:

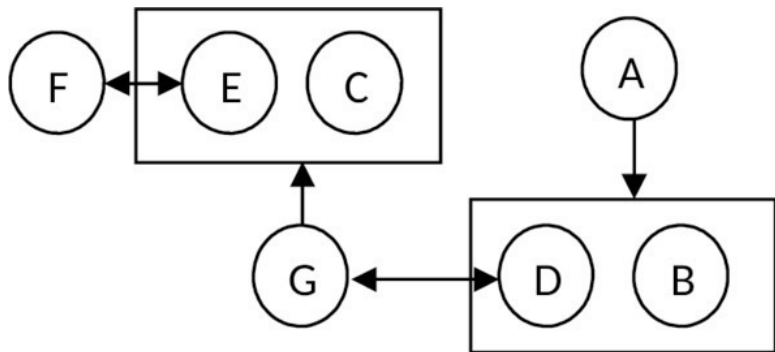
– т. к.  $A \rightarrow D$ ,  $D \rightarrow G$ , то  $A \rightarrow G$  – исключим по аксиоме транзитивности

– т. к.  $G \rightarrow E$ ,  $E \rightarrow F$ , то  $G \rightarrow F$  исключим по аксиоме транзитивности

– т. к.  $G \rightarrow E$ ,  $G \rightarrow C$ , то по аксиоме аддитивности  $G \rightarrow E, C$

– т. к.  $A \rightarrow B$ ;  $A \rightarrow D$ , то по аксиоме аддитивности  $A \rightarrow B, D$

Окончательная диаграмма функциональных зависимостей:



Пример проектирования методом декомпозиции БД интернет-магазин.

Уточнив вопрос о том какую информацию следует хранить в базе данных, определим все атрибуты, представляющие интерес для проектируемой базы данных. Это: для каждого товара его код, название, цена, процентная скидка; ФИО, адрес, телефон каждого клиента; для каждого заказа его код, сумма, дата выполнения и количество товара для каждого заказанного названия.

Применим для всех атрибутов краткие обозначения:

Название товара – НТ

Цена товара – ЦТ

Процентная скидка на товар – ПСТ

Код клиента – КК

ФИО клиента – ФИО

Адрес клиента – АК

Телефон клиента – ТК

Код заказа – КЗ

Сумма заказа – СЗ

Дата выполнения заказа – ДВЗ

Количество товара для каждого заказанного названия – КТЗХ.

Универсальное отношение будет иметь вид: r (НТ, ЦТ, ПСТ, КК, ФИО, АК, ТК, КЗ, СЗ, ДВЗ, КТЗ).

Определив все функциональные зависимости, имеющиеся между атрибутами универсального отношения, построим диаграмму функциональных зависимостей (см. рис. 1.1).

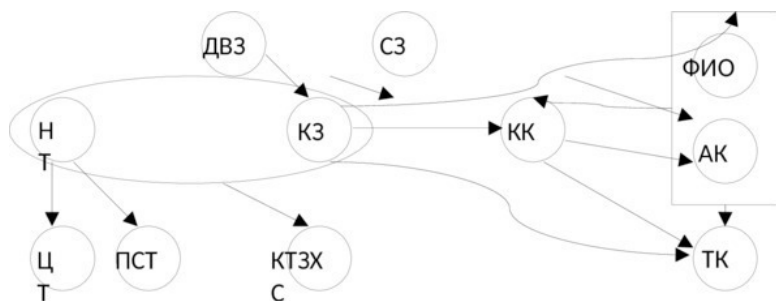


Рис. 1.1. Диаграмма функциональных зависимостей.

Удалим из исходного набора функциональных зависимостей все избыточные:

–  $КЗ \rightarrow КК$  и  $КК \rightarrow ТК$ , т. е.  $КЗ \rightarrow ТК$  также можно удалить по аксиоме транзитивности.

- $КК \rightarrow \Phi ИО$  и  $КК \rightarrow АК$  по аксиоме аддитивности заменим на  $КК \rightarrow \Phi ИО, АК$ .
- $\Phi ИО, АК \rightarrow КК$  и  $КК \rightarrow ТК$ , т. е.  $\Phi ИО, АК \rightarrow ТК$  является избыточной зависимостью по аксиоме транзитивности и ее можно удалить.
- $КЗ \rightarrow КК$  и  $КК \rightarrow \Phi ИО, АК$ , т. е.  $КЗ \rightarrow \Phi ИО, АК$  также можно удалить по аксиоме транзитивности.
- $НТ \rightarrow ЦТ$ ,  $НТ \rightarrow ПСТ$ ,  $НТ \rightarrow НС$  по аксиоме аддитивности заменим на  $НТ \rightarrow ЦТ, ПСТ$ .
- $КЗ \rightarrow СЗ$ ,  $КЗ \rightarrow ДВЗ$ ,  $КЗ \rightarrow КК$  по аксиоме аддитивности заменим на  $КЗ \rightarrow СЗ, ДВЗ, КК$ .
- $КК \rightarrow \Phi ИО, АК$  и  $КК \rightarrow ТК$  по аксиоме аддитивности заменим на  $КК \rightarrow \Phi ИО, АК, ТК$ .

Окончательно диаграмма функциональных зависимостей примет вид, показанный на рис. 1.2.

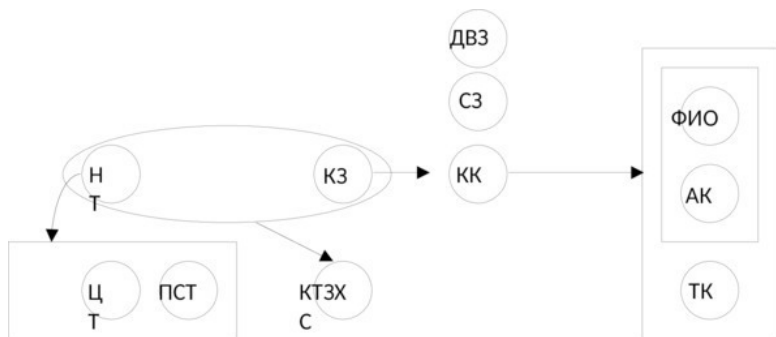


Рис. 1.2. Окончательный вид диаграммы функциональ-

ных зависимостей.

Выполним преобразование исходного отношения в набор НФБК – отношений:

1)

r1 (НТ, ЦТ, ПСТ, КК, ФИО, АК, ТК, КЗ, СЗ, ДВЗ, КТЗ).

Отношение r1 не находится в НФБК (есть зависимости от частей ключа ( $\text{НТ} \rightarrow \text{ЦТ}, \text{ПСТ}$ ); детерминанты НТ, КЗ, КК, (ФИО, АК) не являются возможными ключами) и поэтому разбивается далее.

2) Для проведения проекции по правилу цепочки выберем F-зависимость  $\text{НТ} \rightarrow \text{ЦТ}, \text{ПСТ}$ . Получим следующие отношения:

r2 (НТ, ЦТ, ПСТ);

r3 (НТ, КК, ФИО, АК, ТК, КЗ, СЗ, ДВЗ, КТЗ).

Отношение r2 находится в НФБК (его детерминант (НТ) является возможным ключом) и не нуждается больше в декомпозиции. Отношение r3 не находится в НФБК (есть зависимости от частей ключа ( $\text{КЗ} \rightarrow \text{СЗ}, \text{ДВЗ}, \text{КК}$ ); детерминанты КЗ, КК, (ФИО, АК) не являются возможными ключами) и поэтому разбивается далее.

3) Для проведения второй проекции также по правилу цепочки выберем F-зависимость  $\text{КК} \rightarrow \text{ФИО}, \text{АК}, \text{ТК}$ . Получим следующие отношения:

r4 (КК, ФИО, АК, ТК);

r5 (НТ, КК, КЗ, СЗ, ДВЗ, КТЗ).



Отношение  $r_4$  находится в НФБК (его детерминанты (КК, (ФИО, АК)) являются возможными ключами) и не нуждается больше в декомпозиции. Отношение  $r_5$  не находится в НФБК (есть зависимости от частей ключа ( $K_3 \rightarrow C_3, DV_3, K_3$ )); детерминант  $K_3$  не является возможным ключом) и поэтому разбивается далее.

4) Для проведения третьей проекции по правилу цепочки выберем F-зависимость  $K_3 \rightarrow C_3, DV_3, K_3$ .

$r_6 (K_3, C_3, DV_3, K_3)$ .

Отношение  $r_6$  находится в НФБК (его детерминант ( $K_3$ ) является возможным ключом) и не нуждается больше в декомпозиции.

$r_7 (HT, K_3, KT_3)$ .

Отношение  $r_7$  находится в НФБК (его детерминант (HT,  $K_3$ ) является возможным ключом) и не нуждается больше в декомпозиции.

Преобразование исходного отношения в набор НФБК – отношений завершено.

Таким образом, получили следующий набор отношений:

$r_2 (HT, CT, PST)$ ;

$r_4 (KK, ФИО, АК, ТК)$ ;

$r_6 (K_3, C_3, DV_3, K_3)$ ;

$r_7 (HT, K_3, KT_3)$ .

Выполним проверку полученного набора отношений:

1) Проверим отношения на наличие дублирующихся функциональных зависимостей. Для этого составим списки

F-зависимостей для каждого отношения.

F-зависимости в отношении r2:

HT  $\rightarrow$  ЦТ, ПСТ.

F-зависимости в отношении r4:

КК  $\rightarrow$  ФИО, АК, ТК;

ФИО, АК  $\rightarrow$  КК;

ФИО, АК  $\rightarrow$  ТК.

F-зависимости в отношении r6:

КЗ  $\rightarrow$  СЗ, ДВЗ, КК.

F-зависимости в отношении r7:

HT, КЗ  $\rightarrow$  КТЗ.

Таким образом, в полученном наборе отношений нет F-зависимости, которая появлялась бы более чем в одном отношении. Полученный набор F-зависимостей не совпадает с набором минимального покрытия и может быть получен из него с помощью аксиомы аддитивности (в отношении r4 объединим F-зависимости ФИО, АК  $\rightarrow$  КК и ФИО, АК  $\rightarrow$  ТК).

2) Осуществим проверку набора отношений на наличие избыточных. В полученном наборе отношений нет отношения, все атрибуты которого находились бы в одном другом отношении набора или могли быть найдены в отношении, получаемом с помощью операции соединения любых других отношений проектного набора.

3) Рассмотрим отношения с практической точки зрения. Все полученные отношения разумны с практической точки

зрения: в отношении r2 регистрируются данные о товарах, в отношении r4 хранятся данные о клиентах, отношение r6 отвечает за учет полученных товаров, в отношении r7 записывается информация о количестве каждого товара в заказе.

## **Проектирование базы данных методом «сущность-связь»**

При проектировании базы данных методом «сущность – связь» необходимо выполнить следующие действия:

- Уточнить, какая именно информация о предметной области будет храниться в проектируемой базе данных. Выделить в предметной области объекты и их свойства. Зафиксировать связи между объектами и их свойствами и связи между объектами разных классов. Построить ER – модель.
- Осуществить переход от инфологической модели предметной области к даталогической модели базы данных.
- Выявить, в какой нормальной форме находятся полученные отношения (отобразить функциональные зависимости между атрибутами каждого отношения).

Выделяют три этапа проектирования БД:

- инфологическое моделирование
- даталогическое моделирование
- физическая реализация

1) На первом этапе создается инфологическая модель предметной области.

Предметная область – это часть реального мира, представляющего интерес для данного проектирования.

Инфологической моделью предметной области называют описание предметной области, выполненное с использованием специальных языковых средств, и независимое от используемых в дальнейшем программных и технических средств.

2) На основе инфологической модели строится даталогическая модель.

Даталогическая модель является моделью логического уровня и представляет собой отображение логических связей между элементами данных безотносительно к их содержанию и среде хранения. Описание логической структуры БД на языке СУБД называется **схемой**.

3) Третий этап проектирования состоит в привязке ДЛМ к среде хранения с помощью модели данных физического уровня (**физической модели**). Описание физической структуры БД называется **схемой хранения**.

Компоненты ИЛМ:

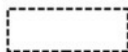
- описание объектов и связей между ними (ER – модель);
- описание информационных потребностей пользователей;
- алгоритмические связи показателей;
- лингвистические отношения;
- ограничения целостности.

В предметной области в результате ее анализа выделяют классы объектов. **Классом объектов** называют совокупность объектов, обладающих одинаковым набором свойств. Каждый объект в информационной системе представляется

своим идентификатором, а каждый класс объектов представляется именем класса. Каждый объект обладает определенным набором свойств.



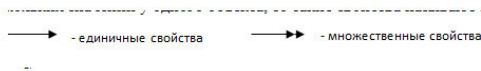
- объект



- свойство

Связь между объектом и характеризующим его свойством изображается в виде линии. Связь может быть единичной или множественной.

Если объект обладает только одним значением какого-то свойства, то такое свойство называют **единичными**. Если для свойства возможно существование одновременно нескольких значений у одного объекта, то такие свойства называют **множественными**.



Свойства, значения которых не могут изменяться с течением времени (например, *Дата рождения*), называются **статическими** и обозначаются буквой *S*. Свойства, значения которых могут изменяться со временем (например, *Фамилия*, *Адрес*, *Телефон*), называются **динамическими** и обозначаются буквой *D*.

Свойство, которое может отсутствовать у некоторых объектов одного класса (например, свойство *Ученая степень*, не все объекты класса *Сотрудники* могут обладать указанным свойством), называют **условными** и изображают пунктирной линией.

Существует понятие **составного** свойства (примеры таких свойств: *Адрес*, состоящий из «улицы», «дома», «квартиры»; *Дата рождения*, состоящая из «числа», «месяца», «года»). Для его обозначения используют квадрат.



### Рис. 3.3. Изображение класса объектов и его свойств

В инфологической модели фиксируются не только связи между объектом и его свойствами, но и связи между объектами разных классов.

Различают связи типа:

- один к одному (1:1);
- один ко многим (1:M);
- многие к одному (M:1);
- многие ко многим (M:M).

Объект называют **простым**, если он рассматривается как неделимый. **Сложный** объект представляет собой объединение других объектов, простых или сложных, также отображаемых в информационной системе. Понятия *простой* и *сложный* являются относительными. Сложные объекты подразделяют на составные, обобщенные и агрегированные.

**Составной** объект соответствует отображению связи «целое – часть». Примеры таких объектов: класс – ученики, группа – студенты и т. п. (связь между составным и составляющими его объектами отображается отношением 1:M)

**Обобщенный** объект отражает наличие связи «род – вид» между объектами предметной области. Например, объекты *Студент*, *Школьник*, *Аспирант* образуют обобщенный объект *Учащиеся*. Объекты, составляющие обобщенный объект, называются его категориями. Как «родовой» объект, так и «видовые» объекты могут обладать определен-

ным набором свойств. Причем «видовые» объекты обладают всеми теми свойствами, которыми обладает «родовой» объект, плюс свойствами, присущими только объектам этого вида.

Определение родо-видовых связей означает классификацию объектов предметной области по тем или иным признакам. Подклассы могут выделяться в ИЛМ в явном виде (см. рис. 3.5).

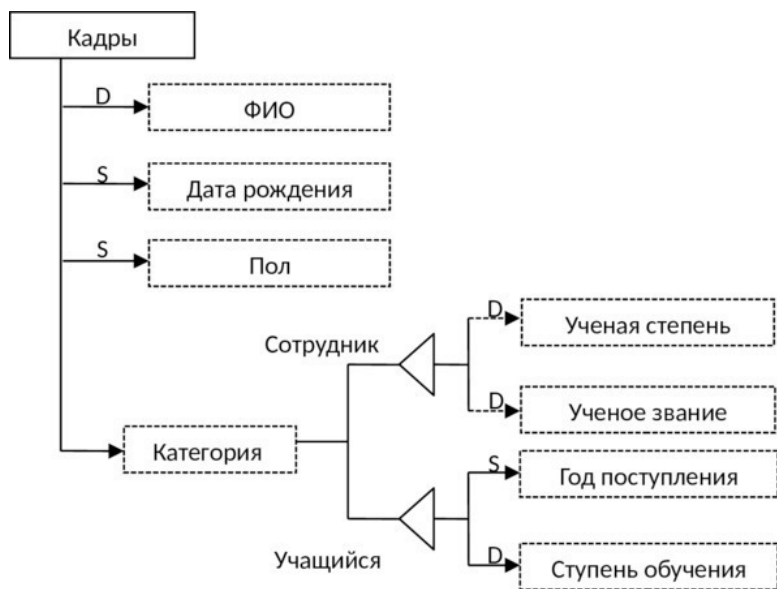


Рис. 3.5. Изображение обобщенного объекта



**Агрегированный** объект соответствует обычно какому-либо процессу, в который оказываются «вовлеченными» другие объекты. Например, агрегированный объект *Поставка* (см. рис. 3.6) объединяет в себе объекты *Поставщик*, *Получатель*, *Продукт* и *Дата*. Для отображения агрегированного объекта в схеме использован ромб. Агрегированный объект может, так же как и простой объект, иметь характеризующие его свойства.

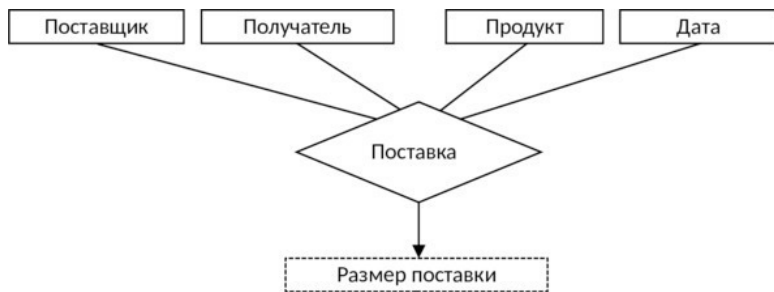
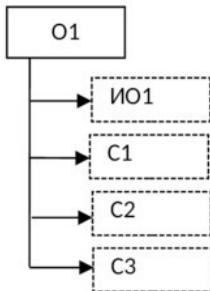


Рис. 3.6. Изображение агрегированного объекта

Правила, по которым строится даталогическая модель:

1) Для каждого простого объекта и его единичных свойств строится таблица, атрибутами которой являются идентификатор объекта и реквизиты, соответствующие каждому из единичных свойств:

Инфологическая  
конструкция

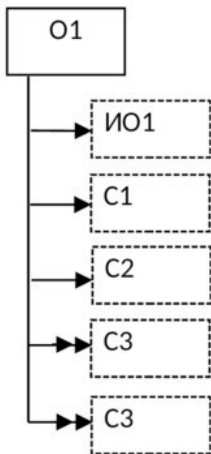


Реляционная схема

$R1(ИО1, C1, C2, C3)$

2) Если у объекта имеются *множественные свойства*, то каждому из них ставится в соответствие отдельная таблица:

Инфологическая  
конструкция



Реляционная схема

R1(ИО1, C1, C2)

R2(ИО1, C3)

R3(ИО1, C4)

3) Если между объектом и его свойством имеется ***условная связь***

# Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.