

Введение в базы данных

Карпук Анатолий Алексеевич,
доцент кафедры ПОСТ
e-mail: A_Karpuk@mail.ru

Лекция 1.

Понятие базы данных. Проектирование логической структуры базы данных

Вопросы:

1. Виды обеспечения АИС
2. Состав информационного обеспечения АИС
3. Организация внутримашинной информационной базы АИС
4. Концепция базы данных
5. Модели данных в СУБД
6. Реляционная модель данных
7. Классификация СУБД
8. Администрирование баз данных
9. Проектирование баз данных. Построение ER-диаграммы
0. Преобразование ER-диаграммы в схему БД
1. Нормализация таблиц базы данных

1. Виды обеспечения АИС

Обеспечивающие подсистемы АИС:

- организационное обеспечение;
- информационное обеспечение;
- техническое обеспечение;
- математическое обеспечение;
- программное обеспечение.

Организационное обеспечение АИС

Организационное обеспечение АИС регламентирует организационную структуру управления объектом, действия персонала по обеспечению функционирования АИС, а также устанавливает функции, права и обязанности должностных лиц по обеспечению функционирования АИС

Информационное обеспечение АИС

Информационное обеспечение регламентирует потоки и подготовку информации, организацию информационной базы, систему классификации и кодирования, определяет технологический процесс обработки информации в АИС

Техническое обеспечение АИС

Техническое обеспечение включает технику автоматизации выполнения информационных процессов в АИС, а также инструкции по ее эксплуатации и обеспечению надежного функционирования.

Состав технического обеспечения определяется организационной структурой управления и информационным обеспечением

Математическое и программное обеспечение АИС

Математическое обеспечение включает **модели, методы и алгоритмы**, применяемые для автоматизации функций управления и выполнения информационных процессов в АИС. Программное обеспечение представляет собой **комплексы программ, реализующих методы и алгоритмы математического обеспечения.**

Состав математического и программного обеспечения зависит от **автоматизируемых функций управления и информационного обеспечения АИС**

2. Состав информационного обеспечений АИС

В АИС реализуются функции **информационной системы**. Информационная система осуществляет **фиксацию информации** о состоянии объекта управления и **внешних возмущениях**, организацию полученных данных и снабжение необходимыми данными управляющей системы.

Организация данных включает **представление данных и управление данными** в соответствии с определенными соглашениями. Управление данными представляет собой совокупность функций **обеспечения требуемого представления данных, их накопления, хранения, обновления, удаления, поиска по заданным критериям и выдачи.**

Состав информационного обеспечения АИС

Подсистемы:

- **сбора и передачи информации;**
- **классификации и кодирования;**
- **организации внемашинной информационной базы;**
- **организации внутримашинной информационной базы**

Внемашинную информационную базу образует совокупность всех документированных данных и сообщений используемых в АИС.

Внутримашинную информационную базу образует совокупность всех данных на машинных носителях, сгруппированных по определенному признаку.

3. Организация внутримашинной информационной базы АИС

Включает:

- логическую организацию данных;
- физическую организацию данных;
- методы доступа к данным;
- средства ведения данных;
- методы и средства обеспечения целостности и достоверности данных;
- методы и средства защиты данных.

Организация внутримашинной информационной базы АИС

Логическая организация данных учитывает лишь те конструкции данных и операции над ними, которые находятся в распоряжении программы, использующей данные.

Физическая организация данных учитывает размещение и связи данных в среде хранения.

Методы доступа к данным – это совокупность соглашений и средств, с помощью которых реализуется заданный вид доступа к физическим записям.

Организация внутримашинной информационной базы АИС

Ведением информационной базы называется деятельность по обновлению, восстановлению и перестройке ее структуры с целью обеспечения целостности, сохранности и эффективности использования данных.

Методы и средства обеспечения целостности и достоверности данных обеспечивают корректность и непротиворечивость данных при выполнении операций над данными.

Защита данных включает организационные, программные и технические методы и средства, направленные на удовлетворение ограничений по доступу к данным.

Организация внутримашинной информационной базы АИС



Организация внутримашинной информационной базы АИС

Недостатки традиционной организации данных – многократное дублирование данных и невозможность оперативного внесения изменений в группы данных, дублированные в нескольких файлах.

Недостатки метода единой информационной базы – включение в АИС новой ПЗ, не предусмотренной при проектировании информационной базы, может привести к ее изменению и к изменению программ разработанных ПЗ.

4. Концепция базы данных

Базой данных (БД) называется совокупность данных, организованная по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ.

Системой управления базами данных (СУБД) называется совокупность программ и языковых средств, предназначенных для управления данными в БД, ведения БД и обеспечения взаимодействия ее с прикладными программами.

Модели данных и схемы БД

Модель данных – это совокупность правил порождения структур данных в БД, операций над данными, а также ограничений целостности, определяющих последовательность изменения, допустимые связи и допустимые значения данных.

Концепция БД предусматривает три уровня описания данных: **внешний, концептуальный и внутренний**. На каждом уровне используется соответствующая модель данных.

Описание БД в контексте конкретной модели данных называется **схемой БД**.

Модели данных и схемы БД

Внешние схемы БД применяются для описания данных в виде, используемом программами ПЗ.

Концептуальная схема БД определяет представление БД, единое для всех ПЗ и не зависящее от используемого в СУБД представления данных в среде хранения и путей доступа к ним.

Внутренняя схема БД определяет представление данных в среде хранения и пути доступа к ним.

Внешние и концептуальная схема относятся к **логической организации данных**, а внутренняя схема – к **физической организации данных**.

Языковые средства СУБД

При описании БД внешние, **концептуальная и внутренняя схемы БД** описываются на **языке описания данных (ЯОД)**, входящем в состав языковых средств СУБД.

Каждая ПЗ формирует запросы на добавление, удаление, обновление и поиск данных в БД в соответствии со своей внешней схемой на **языке манипулирования данными (ЯМД)**.

Возможности СУБД

СУБД обеспечивают:

- **уменьшение избыточности** хранимых данных за счет минимизации дублирования данных;
- **достоверность** хранимых данных за счет автоматической корректировки всех дублируемых элементов данных;
- **стандартизацию** данных в различных ПЗ;
- **совместное использование** хранимых данных различными ПЗ в мультипрограммном режиме;

СУБД обеспечивают:

- **физическую независимость данных**, заключающуюся в возможности модификации внутренней схемы БД без изменения концептуальной схемы БД, внешних схем и программ ПЗ;
- **логическую независимость данных**, заключающуюся в возможности модификации концептуальной схемы БД без изменения внешних схем и программ ПЗ;

СУБД обеспечивают:

- **целостность данных** за счет проверки ограничений целостности при добавлении, удалении и обновлении данных, связанных изменений данных и механизма обработки транзакций;
- **разграничение доступа к данным** за счет защиты данных СУБД и за счет описания во внешней схеме каждой ПЗ только тех данных, которые она использует;
- **сохранность данных** при возникновении сбоев и отказов сервера за счет средств резервного копирования и восстановления БД.

5. Модели данных в СУБД

Модель данных – это совокупность правил порождения структур данных в базе данных, операций над ними, а также ограничений целостности, определяющих допустимые связи и значения данных, последовательность их изменения [ГОСТ 20886-85].

Модель данных состоит из трёх частей:

1. Набор типов структур данных.

Здесь можно провести аналогию с языками программирования, в которых тоже есть predefined типы структур данных, такие как скалярные данные, векторы, массивы, структуры (например, тип *struct* в языке Си) и т.д.

2. Набор операторов или правил вывода, которые могут быть применены к любым правильным примерам типов данных, перечисленных в (1), чтобы находить, выводить или преобразовывать информацию, содержащуюся в любых частях этих структур в любых комбинациях.

3. Набор общих правил целостности, которые прямо или косвенно определяют множество непротиворечивых состояний базы данных и/или множество изменений её состояния.

Операции над данными

Модель данных определяет множество действий, которые допустимо производить над некоторой реализацией БД для её перевода из одного состояния в другое. Это множество соотносят с **языком манипулирования данными** (Data Manipulation Language, DML).

Любая операция над данными включает в себя **селекцию данных** (select). **Условие селекции** – это некоторый критерий отбора данных, в котором могут быть использованы логическая позиция элемента данных, его значение и связи между данными.

По типу производимых действий различают следующие операции:

- идентификация данных и нахождение их позиции в БД;
- выборка (чтение) данных из БД;
- включение (запись) данных в БД;
- удаление данных из БД;
- модификация (изменение) данных БД.

Обработка данных в БД осуществляется с помощью процедур базы данных – транзакций. **Транзакцией** называют упорядоченное множество операций, переводящих БД из одного согласованного состояния в другое.

Ограничения целостности

Ограничения целостности – это правила, которым должны удовлетворять значения элементов данных. Ограничения целостности делятся на:

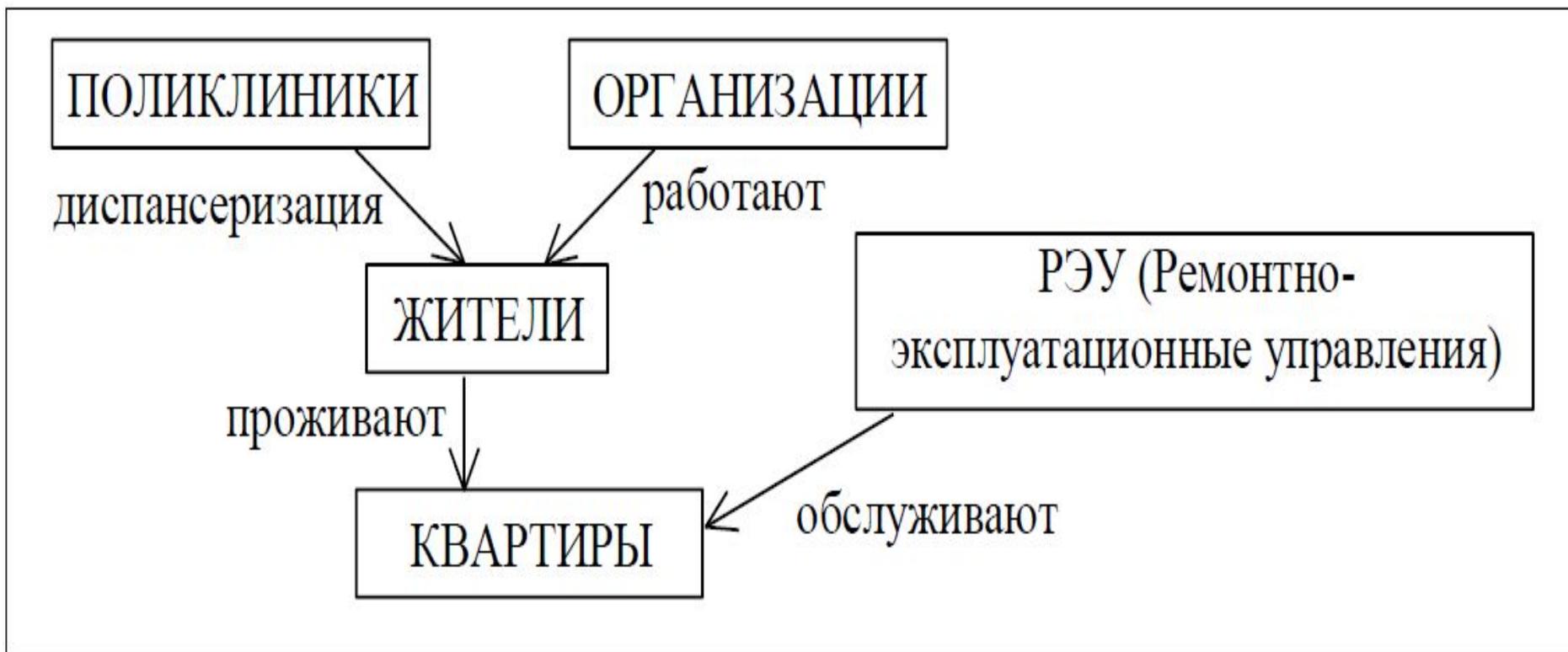
- **явные** (включаются в структуру базы данных с помощью средств языка контроля данных (DCL, Data Control Language))
- **неявные** (определяются самой структурой данных).

Также различают **статические** и **динамические** ограничения целостности. Статические ограничения присущи всем состояниям ПО, а динамические определяют возможность перехода ПО из одного состояния в другое.

За выполнением ограничений целостности следит СУБД в процессе своего функционирования. Она проверяет ограничения целостности каждый раз, когда они могут быть нарушены (например, при добавлении данных, при удалении данных и т.п.), и гарантирует их соблюдение.

Таким образом, ограничения целостности обеспечивают логическую непротиворечивость данных при переводе БД из одного состояния в другое.

Пример структуры БД сетевой модели данных



Операции над данными в сетевой модели данных

В СМД применяются следующие операции над данными:

- *запомнить*: внесение информации в БД;
- *включить в групповое отношение*: установление связей между данными;
- *переключить*: переход члена набора к другому владельцу;
- *обновить*: модификация данных;
- *извлечь*: чтение данных;
- *удалить*: физическое или логическое удаление данных;
- *исключить из группового отношения*: разрыв связей между данными.

Иерархическая модель данных

Графическая диаграмма концептуальной схемы базы данных называется *деревом определения*. Пример иерархической базы данных:



ИМД не
поддерживает:

**необязательный класс членства и
ручной режим включения записей.**

6. Реляционная модель данных (РМД)

- В 1970 г. американский математик Э.Ф.Кодд опубликовал статью, с которой отсчитывается начало существования РМД.
- РМД основана на теории множеств.
- Домен, D – множество значений, которые может принимать элемент данных.
- Декартово произведение доменов – множество всех возможных комбинаций значений доменов:
 $D_1 \times D_2 \times \dots \times D_n = \{(d_{1i}, d_{2i}, \dots, d_{ni})\}$, где $d_{ki} \in D_k$
- Пример: $D_1 = (1, 2)$, $D_2 = (a, b, c)$.
 $D_1 \times D_2 = \{(1,a), (1,b), (1,c), (2,a), (2,b), (2,c)\}$
- Отношение – подмножество декартова произведения доменов.

Пример таблицы реляционной БД

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<i>Год рождения</i>	<i>Отдел</i>
023	Волкова Елена Павловна	секретарь	26000	1985	2
113	Белов Сергей Юрьевич	инженер	39800	1980	1
101	Рогов Сергей Михайлович	директор	62000	1972	2
056	Панина Анна Алексеевна	инженер-программист	41800	1978	1
...
098	Фролов Юрий Вадимович	начальник отдела	49200	1971	9

Мощность отношения (кардинальность) – количество кортежей.

Степень отношения (арность) – количество атрибутов.

Термины. Свойства отношения

первичный ключ

Отношение, таблица

столбец, поле, атрибут

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<i>Год рождения</i>	
023	Волкова Елена Павловна	секретарь	26000	1985	← описание (схема отношения)
113	Белов Сергей Юрьевич	инженер	39800	1980	← строка, запись, кортеж

Отношение обладает двумя основными свойствами:

1. В отношении не должно быть одинаковых кортежей, т.к. это множество.
2. Порядок кортежей в отношении несущественен.

Ключи отношения

- **Ключ** – атрибут (группа атрибутов), которые позволяют классифицировать кортеж (запись таблицы).
- **Потенциальный ключ (уникальный ключ)** – атрибут (группа атрибутов), которые позволяют идентифицировать кортеж (запись таблицы).
- **Первичный ключ** – обязательный уникальный ключ. Для каждой таблицы может быть определен только один первичный ключ.
- **Вторичный ключ** – любой другой ключ, кроме первичного. Может быть необязательным и неуникальным.
- **Внешний ключ** – служит для организации связей между таблицами.

Организация связей между таблицами

Связь один-ко-многим: Отделы – Сотрудники

Таблица «Сотрудники»

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Отдел</i>
023	Волкова Елена Павловна	2
113	Белов Сергей Юрьевич	1
101	Рогов Сергей Михайлович	2
056	Панина Анна Алексеевна	1
...
098	Фролов Юрий Вадимович	9

Таблица «Отделы»

<i>Номер отдела</i>	<i>Название отдела</i>
1	Информационный отдел
2	Администрация
3	Отдел кадров
...	...
9	Проектный отдел

«Номер отдела» - первичный ключ в таблице «Отделы»

«Отдел» – внешний ключ в таблице «Сотрудники».

Внешний ключ – ограничение целостности, в соответствии с которым множество значений внешнего ключа является подмножеством значений первичного или уникального ключа этой же или другой таблицы.

Организация связей между таблицами

Связь многие-ко-многим:
Проекты – Сотрудники



Таблица «Сотрудники»

<i>ФИО</i>	<i>Номер</i>
Волкова Е.П.	023
Белов С.Ю.	113
Рогов С.М.	101
Панина А.А.	056
Фролов Ю.В.	098
...	...

Таблица «Участие»

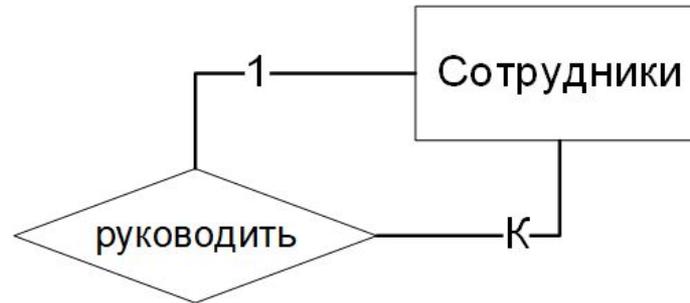
<u><i>Участник</i></u>	<i>Роль</i>	<u><i>Проект</i></u>
113	исполнитель	23/Н
101	руководитель	18-К
056	исполнитель	18-К
101	консультант	09/Р
098	руководитель	23/Н
...

Таблица «Проекты»

<i>Шифр</i>	<i>Название проекта</i>
23/Н	АИС "Налог"-2
18-К	ИПС "Жители"
09/Р	ГИС "Город"

В таблице «Участие»: «Участник» – внешний ключ к таблице «Сотрудники»
«Проект» – внешний ключ к таблице «Проекты»

Пример связи внутри таблицы



<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<u><i>Начальник</i></u>
023	Волкова Елена Павловна	секретарь	26000	101
113	Белов Сергей Юрьевич	инженер	39800	205
101	Рогов Сергей Михайлович	директор	62000	NULL
205	Махова Ольга Алексеевна	начальник отдела	51300	101
...

Операции над данными в РМД

Операции применяются к кортежам отношений.

В РМД используются следующие операции:

- **запомнить**: внесение информации в БД (требует формирования значений уникального ключа и обязательных атрибутов кортежа);
- **извлечь**: чтение данных;
- **обновить**: модификация данных – изменение значений атрибутов кортежей;
- **удалить**: физическое или логическое удаление данных (кортежей).

При наличии внешних ключей операции модификации данных также отвечают за установление / изменение / разрыв связей.

Объектно-реляционная модель данных

Объектно-реляционная модель данных (ОРМД) реализована с помощью реляционных таблиц, но включает объекты, аналогичные объектам в объектно-ориентированном программировании. В ОРМД используются такие объектно-ориентированные компоненты, как пользовательские типы данных, инкапсуляция, полиморфизм, наследование, переопределение методов и т. п.

В большинстве реализаций ОРМД объектами признаются агрегат и таблица (отношение), которая может входить в состав другой таблицы. Методы обработки данных представлены в виде хранимых процедур и триггеров, которые являются процедурными объектами базы данных и связаны с таблицами. На концептуальном уровне все данные объектно-реляционной БД представлены в виде отношений, и ОРСУБД поддерживают язык SQL.

7. Классификация СУБД

По степени универсальности СУБД делят на два класса:

1. СУБД общего назначения (СУБД ОН)
2. специализированные СУБД (СпСУБД).

Специализированные СУБД создаются в тех случаях, когда ни одна из существующих СУБД общего назначения не может удовлетворительно решить задачи, стоящие перед разработчиками. Причин может быть несколько:

- не достигается требуемого быстродействия обработки данных;
- необходима работа СУБД в условиях жёстких аппаратных ограничений;
- требуется поддержка специфических функций обработки данных.

СпСУБД предназначены для решения конкретной задачи, а приемлемые параметры этого решения достигаются следующим образом:

1. за счёт знания особенностей конкретной предметной области,
2. путём сокращения функциональной полноты системы.

Классификация СУБД

По методам организации хранения и обработки данных СУБД делят на

□ **Централизованные**

□ **Распределённые.**

Первые работают с БД, которая физически хранится в одном месте (на одном компьютере). Это не означает, что пользователь может работать с БД только за этим же компьютером: доступ может быть удалённым (в режиме клиент–сервер). Большинство централизованных СУБД перекладывает задачу организации удалённого доступа к данным на сетевое обеспечение, выполняя только свои стандартные функции, которые усложняются за счёт одновременности доступа многих пользователей к данным.

По модели данных различают **иерархические, сетевые, реляционные, объектно-реляционные и объектно-ориентированные СУБД.**

Классификация СУБД

По месту запуска ядра СУБД делятся на **локальные и клиент-серверные**. В локальных СУБД ядро запускается на локальном компьютере, на котором выполняются приложения, взаимодействующие с СУБД, и за которым работает пользователь, взаимодействующий с СУБД.

К локальным СУБД относятся **dBase, FoxPro, Paradox, Microsoft Access**.

В клиент-серверных СУБД ядро запускается на сервере, приложения пользователи взаимодействуют с СУБД через локальную или глобальную сеть с клиентских компьютеров. К клиент-серверным СУБД относятся **Firebird, Interbase, IBM DB2, Microsoft SQL Server, Sybase, Oracle, PostgreSQL, MySQL**

8. Администрирование баз данных

Основные задачи администрирования базы данных – обеспечение надежного и эффективного функционирования системы БД, адекватности содержания БД информационным потребностям пользователей, отображения в БД актуального состояния ПрО.

Администрирование БД возлагается на администратора (или персонал администрирования, если АИС с БД достаточно крупная). В задачи администратора БД входит выполнение нескольких групп функций:

1. Администрирование предметной области: поддержка представления БД на концептуальном уровне архитектуры СУБД (общем для всех приложений); адекватное отображение в БД изменений, происходящих в ПрО. Последнее требование может подразумевать реструктуризацию (изменение схемы) БД и последующее приведение содержимого БД в соответствие с новой схемой.

Администрирование баз данных

2. Администрирование БД: поддержка представления БД в среде хранения, эффективная и надежная эксплуатация системы БД. Если на этом уровне проводится реорганизация БД (с целью повышения эффективности работы), то она заключается в следующем:

- изменения в структуре хранимых данных, например, выведение в отдельную таблицу редко используемых данных;

- изменения способов размещения данных в памяти, например:

 - разбиение таблицы на части для распределения её по различным физическим носителям с целью распараллеливания доступа к ней;

 - разбиение БД в среде хранения на кластеры;

 - изменение физических параметров среды хранения, например,

 - размера блока данных в пространстве памяти.

- изменения используемых методов доступа к данным, например, построение индексов или введение хеширования.

Администрирование баз данных

3. Администрирование приложений: поддержка представлений БД для различных групп пользователей механизмами внешнего уровня СУБД. При изменении концептуальной схемы БД или схемы хранения может потребоваться внесение соответствующих изменений во внешние схемы.

4. Администрирование безопасности данных: предоставление пользователям прав на доступ к БД и настройка системных средств защиты от несанкционированного доступа.

В состав СУБД обычно включаются вспомогательные средства (различные утилиты), упрощающие администрирование БД.

9. Проектирование баз данных.

Построение ER-диаграммы

- I. Информационно-логическое (инфологическое) проектирование
 - анализ предметной области;
 - построение модели предметной области;
 - определение границ информационной поддержки;
 - определение групп пользователей.
- II. Определение требований к операционной обстановке:
 - выбор аппаратной платформы;
 - выбор операционной системы.
- III. Выбор СУБД и других инструментальных программных средств.
 - выбор СУБД;
 - выбор версии СУБД и архитектуры, в которой она будет работать.
- IV. Логическое проектирование БД (даталогическое):
 - преобразование схемы предметной области в схему базы данных;
 - нормализация и создание схем отношений.
- V. Физическое проектирование БД:
 - разработка физической структуры БД (внутренней схемы);
 - реализация проекта на DDL-языке выбранной СУБД;
 - создание дополнительных объектов БД (индексов, представлений, триггеров и др.).

Понятие предметной области

- Под **предметной областью** понимается совокупность объектов (одушевленных и неодушевленных предметов, групп предметов, явлений, событий, процессов) реального мира, значения свойств (характеристик, атрибутов) которых будут храниться и обрабатываться в базе данных.

Методология проектирования базы данных



Модель данных «сущность-связь»

Для построения **инфологической модели предметной области** используются **диаграммы «сущность-связь» (ER-диаграммы)**.

Множество **допустимых структурных компонентов** модели данных «сущность-связь»:

- сущность;
- связь между сущностями;
- атрибут сущности;
- первичный ключ сущности;
- уникальный ключ сущности;
- внешний ключ сущности;
- функциональные зависимости (ФЗ) между атрибутами сущности;
- состав многозначного атрибута сущности;
- ФЗ между элементами многозначного атрибута сущности.

Сущности в модели данных

«СУЩНОСТЬ-СВЯЗЬ»

Сущность – это множество **реальных** или **абстрактных объектов** (людей, предметов, документов и т.п.), **обладающих общими атрибутами или характеристиками**. Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована. Именование сущности осуществляется с помощью существительного в единственном числе. При этом имя сущности должно отражать **тип** или **класс** объекта, а не его **конкретный экземпляр**.

Сущности не существуют отдельно друг от друга. Между ними имеются **отношения**, которые должны быть отражены в инфологической модели предметной области в виде **связей**.

Связь представляет собой **соединение двух сущностей**. Связь описывается **вербальными фразами** в виде глаголов в двух направлениях. Каждая связь должна иметь свое **уникальное имя связи**.

Атрибуты сущностей

Атрибут сущности – свойство сущности, имеющее **имя**, **область допустимых значений** (тип и формат) и **признак обязательности** атрибута. Значением атрибута может быть **неделимый элемент данных**, **вектор из элементов данных**, **структура из элементов данных**, **повторяющаяся группа из элементов данных**, **векторов** или **структур**.

Сущность должна иметь **обязательный атрибут** или **комбинацию обязательных атрибутов**, чьи значения **однозначно определяют** каждый экземпляр сущности. Эти атрибуты образуют **первичный ключ сущности (Primary Key, PK)**. Если сущность имеет несколько таких подмножеств атрибутов, то одно из них объявляется **первичным ключом**, а каждое из остальных объявляется **уникальным** или **альтернативным ключом**.

Многозначные атрибуты сущностей

Если значением **многозначного атрибута** является **вектор или структура**, то каждому **элементу** вектора или структуры ставится в соответствие **атомарный атрибут**. В инфологической модели ПрО следует задать **имена, типы и форматы этих атомарных атрибутов**.

Если значением **многозначного атрибута** является **повторяющаяся группа** из элементов данных, векторов или структур, то в инфологической модели ПрО следует задать **имена, типы и форматы атомарных атрибутов, входящих в повторяющуюся группу**. Кроме того, среди атомарных атрибутов повторяющейся группы следует выделить **ключевые и не ключевые атрибуты**.

Функциональные зависимости между атрибутами сущностей

Пусть X – некоторое подмножество атрибутов сущности, Y – атрибут сущности, не входящий в X .

Множество атрибутов X функционально определяет атрибут Y (атрибут Y функционально зависит от атрибутов X), обозначается $X \rightarrow Y$, если **любой комбинации** значений атрибутов из X соответствует **одно значение** атрибута Y . Другими словами, для любых двух экземпляров сущности из **равенства значений** атрибутов из X следует **равенство значений** атрибута Y .

В **левой части ФЗ** содержится **один или более** атрибутов. Если в **правой части ФЗ** записано более одного атрибута, то **каждый из этих атрибутов функционально зависит** от левой части ФЗ.

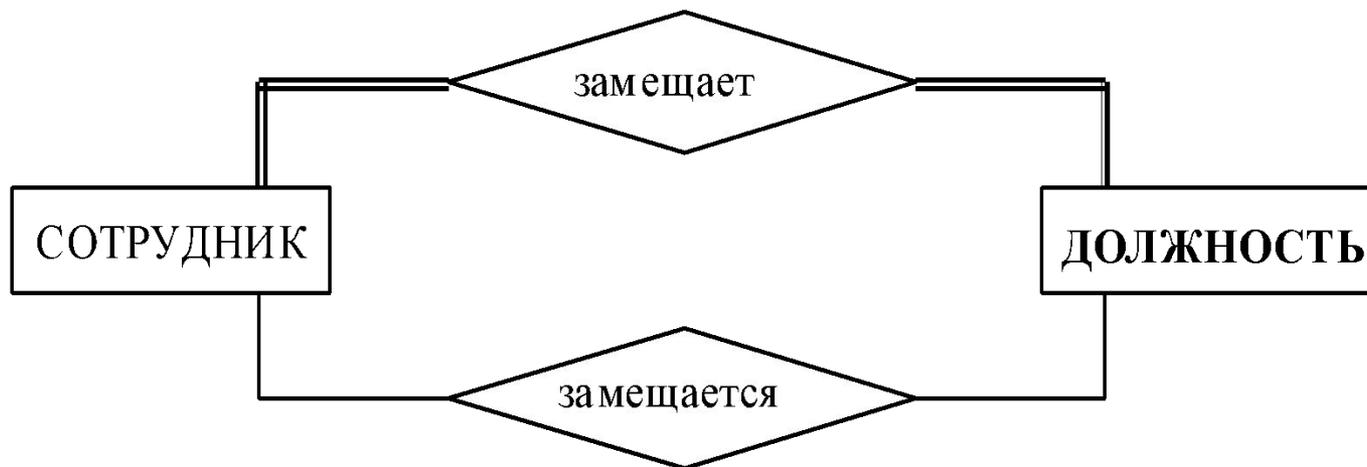
Связи между сущностями

Связи между сущностями:

- ✓ Для связи указывается:
 - название**,
 - тип** (факультативная или обязательная),
 - кардинальность** (1:1, 1:n или m:n),
 - степень** (унарная, бинарная, тернарная или n-арная).

Различают *тип связи* и *экземпляр связи*.

Примеры обязательной и факультативной связей:

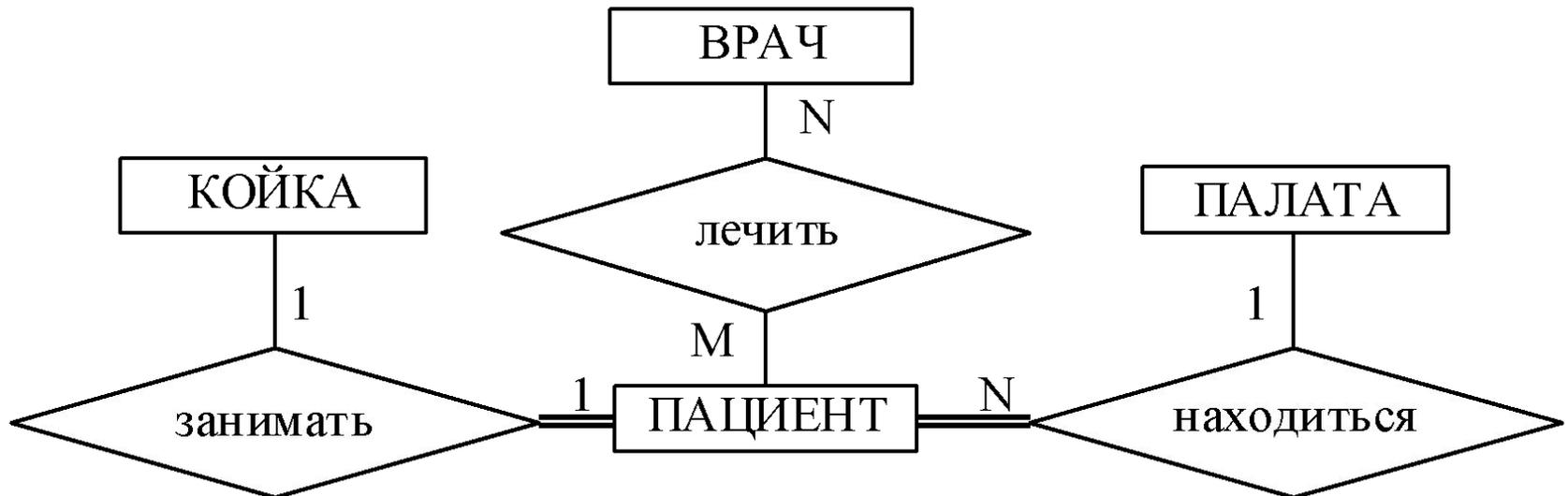


Связи между сущностями

Кардинальность связей между сущностями:

- ✓ один-к-одному (1:1);
- ✓ один-ко-многим (1:n);
- ✓ многие-ко-многим (m:n).

Примеры связей разной кардинальности:



Внешние ключи сущностей

Если между двумя сущностями имеется **связь «один к одному»** или **«один ко многим»** («многие к одному»), то **атрибуты первичного ключа** родительской сущности наследуются в качестве **атрибутов подчиненной сущности**. Эти атрибуты называются **внешним ключом (Foreign Key, FK)**.

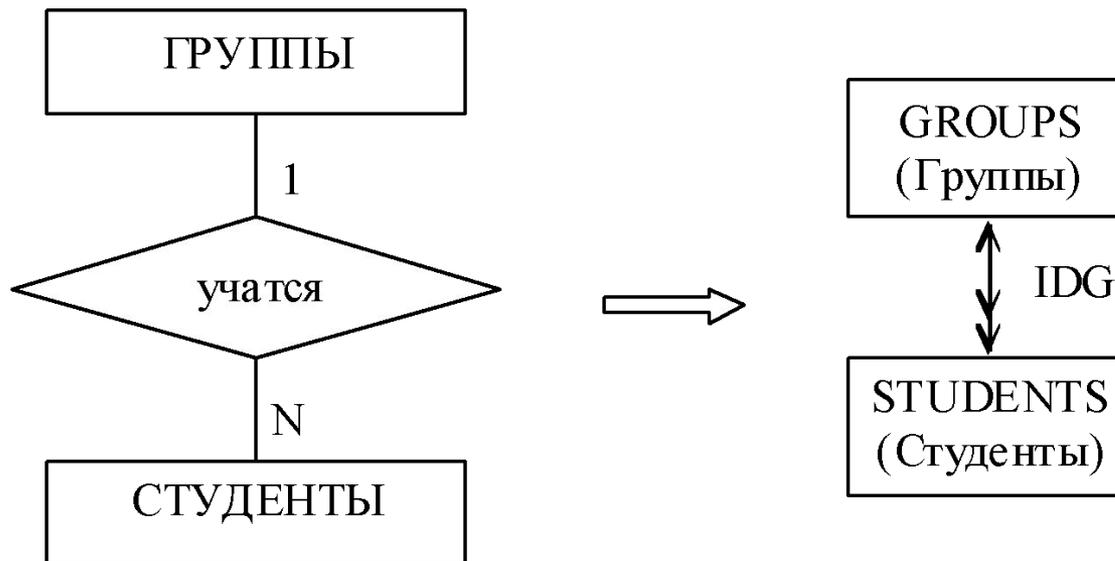
Если связь между сущностями **идентифицирующая**, то атрибуты **внешнего ключа** входят в состав **первичного ключа подчиненной сущности**, либо входят в состав **альтернативного ключа** этой сущности.

Если связь между сущностями **не идентифицирующая**, то атрибуты **внешнего ключа** входят в состав **не ключевых атрибутов подчиненной сущности**.

10. Преобразование ER-диаграммы в схему базы данных

Правила преобразования:

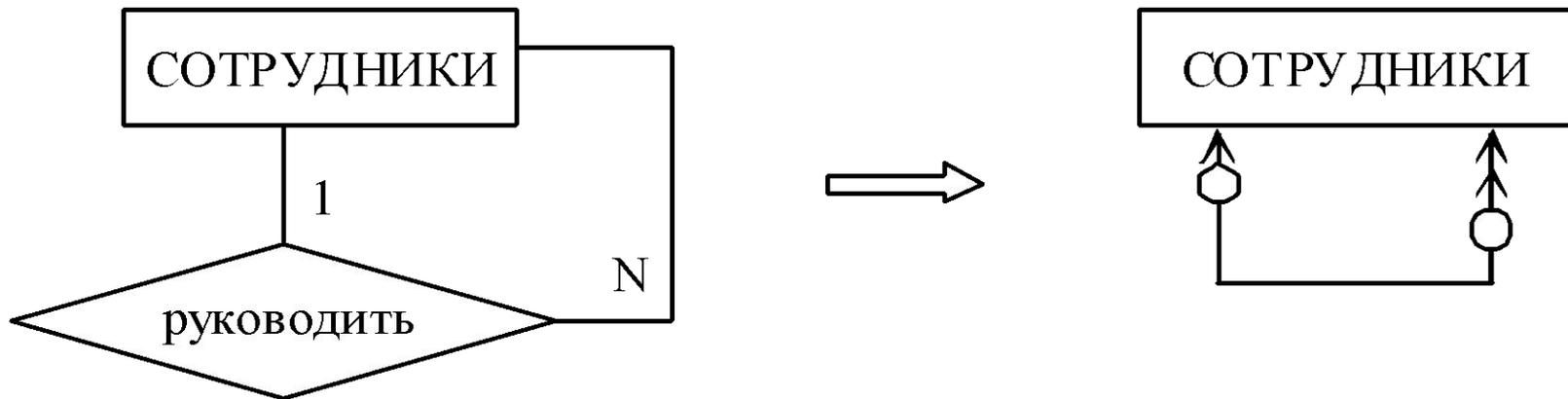
1. Каждый тип сущности преобразуется в таблицу БД. В таблицу вносятся все атрибуты, относящиеся к данному типу сущности.
2. Бинарная связь 1:n (между сущностями **разных типов**) реализуется с помощью внешнего ключа между двумя таблицами



Преобразование ER-диаграммы в схему базы данных

Правила преобразования:

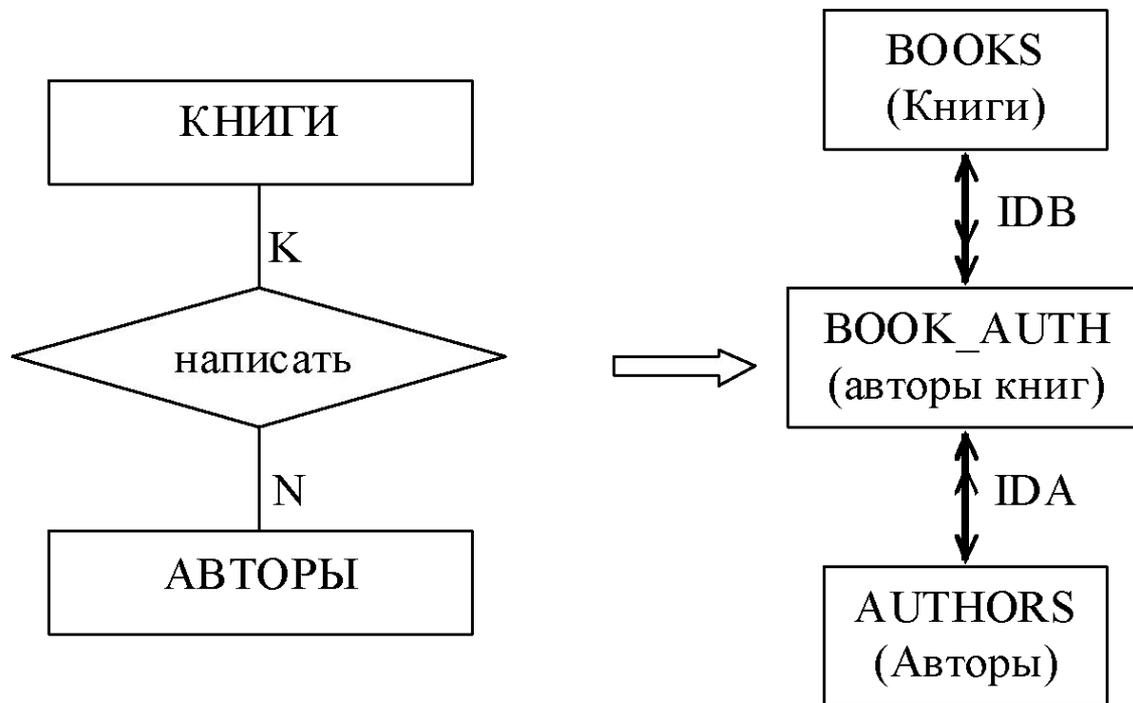
3. Связь 1:1 реализуется в рамках одной таблицы. Исключение из этого правила составляют ситуации, когда связанные сущности существуют независимо друг от друга.
4. Унарная связь 1:n (между сущностями **одного типа**) реализуется с помощью внешнего ключа, определённого в той же таблице, что и первичный ключ.



Преобразование ER-диаграммы в схему базы данных

Правила преобразования:

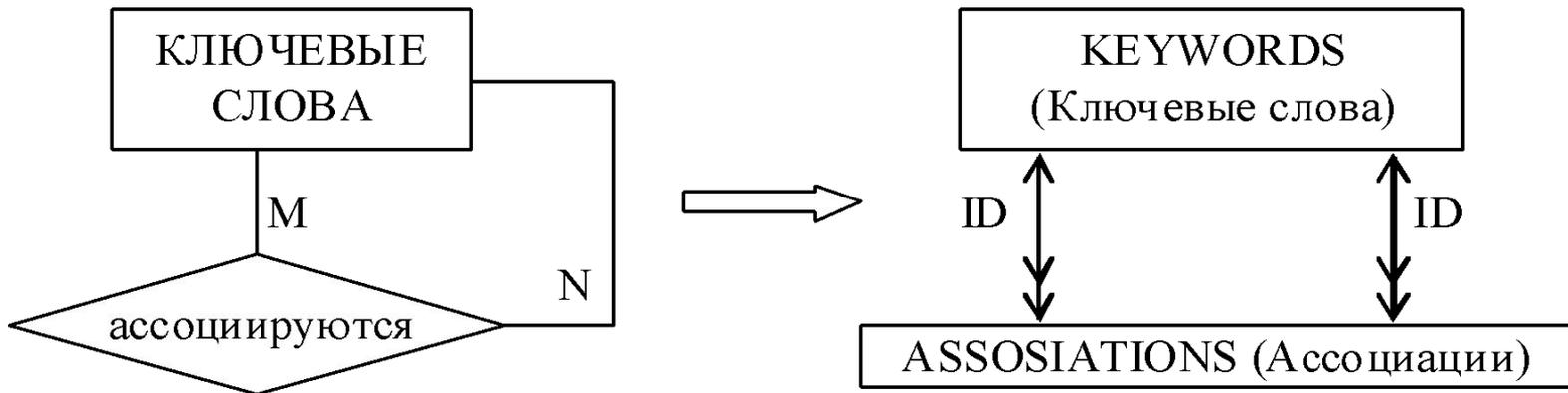
- Бинарная связь типа $n:m$ реализуется с помощью промежуточной таблицы.



Преобразование ER-диаграммы в схему базы данных

Правила преобразования:

6. Унарная связь n:m реализуется с помощью промежуточной таблицы.



11. Нормализация таблиц БД. Аномалии модификации данных

При неправильно спроектированной схеме реляционной БД могут возникнуть аномалии выполнения операций модификации данных.

Рассмотрим эти аномалии на примере следующего отношения:

ПОСТАВКИ (*Номер поставки, Название товара, Цена товара, Количество, Дата поставки, Название поставщика, Адрес поставщика*)

Различают аномалии обновления, удаления и добавления.

Аномалия обновления может возникнуть в том случае, когда информация дублируется. Другие аномалии возникают тогда, когда две и более сущности объединены в одно отношение.

Например:

- **Аномалия обновления:** изменился адрес поставщика. Если от него было несколько поставок, то придется менять несколько записей.
- **Аномалия удаления:** при удалении в архив записей обо всех поставках определённого поставщика все данные об этом поставщике (название, адрес) будут утеряны.
- **Аномалия добавления:** нельзя добавить сведения о поставщике, пока от него нет ни одной поставки.

Для решения проблемы аномалии модификации данных при проектировании РБД проводится нормализация отношений.

Пример для демонстрации нормализации

В таблице приведен пример содержимого исходного отношения КНИГИ:

Таблица 8.1. Исходное отношение КНИГИ

<u>ID</u>	<u>Code</u>	<u>Theme</u>	<u>Author</u>	<u>Title</u>	<u>Editor</u>	<u>Type</u>	<u>Year</u>	<u>Pg</u>
20	22.18	МК	Бочков С.	Язык программирования СИ	Садчиков П.	учебник	1990	384
			Субботин Д.		Седов П.			
10	22.18	МК	<u>Джехани Н.</u>	Язык АДА	Красилов А.	учебник	1988	552
					<u>Перминов О.</u>			
35	32.97	ВТ	Соловьев Г.	Операционные системы ЭВМ		учебное пособие	1992	208
			Никитин В.					
11	32.81	<u>Кибернетика</u>	Попов Э.В.	Общение с ЭВМ на естественном языке	Некрасов А.	учебник	1982	360
44	32.97	ВТ		ПУ для ПЭВМ	<u>Витенберг Э.</u>	<u>справочник</u>	1992	208
89	32.973	ЭВМ	<u>Коутс Р.</u>	Интерфейс «человек-компьютер»	Шаньгин В.	учебник	1990	501
			<u>Влейминк И.</u>					

Первая нормальная форма

Введём понятие простого и сложного атрибута:

Простой атрибут – это атрибут, значения которого атомарны (т.е. неделимы).

Сложный атрибут может иметь значение, представляющее собой конкатенацию нескольких значений одного или разных доменов. Аналогом сложного атрибута может быть агрегат или повторяющийся агрегат данных.

Первая нормальная форма (1НФ).

Таблица приведена к 1НФ, если все ее атрибуты простые.

Приведение к 1НФ

Таблица 8.2. Отношение КНИГИ, приведённое к 1НФ

<u>ID</u>	<u>Code</u>	<u>Theme</u>	<u>Author</u>	<u>Title</u>	<u>Editor</u>	<u>Type</u>	<u>Year</u>	<u>Pg</u>
20	22.18	МК	Бочков С.	Язык программирования СИ	Садчиков П.	учебник	1990	384
20	22.18	МК	Субботин Д.	Язык программирования СИ	Седов П.	учебник	1990	384
10	22.18	МК	<u>Джехани Н.</u>	Язык АДА	Красилов А.	учебник	1988	552
10	22.18	МК	<u>Джехани Н.</u>	Язык АДА	<u>Перминов О.</u>	учебник	1988	552
35	32.97	ВТ	Соловьев Г.	Операционные системы ЭВМ		учебное пособие	1992	208
35	32.97	ВТ	Никитин В.	Операционные системы ЭВМ		учебное пособие	1992	208
11	32.81	<u>Кибер-нетика</u>	Попов Э.В.	Общение с ЭВМ на естественном языке	Некрасов А.	учебник	1982	360
44	32.97	ВТ		ПУ для ПЭВМ	<u>Витенберг Э.</u>	<u>справочник</u>	1992	208
89	32.973	ЭВМ	<u>Коутс Р.</u>	Интерфейс «человек-компьютер»	Шаньгин В.	учебник	1990	501
89	32.973	ЭВМ	<u>Влейминк И.</u>	Интерфейс «человек-компьютер»	Шаньгин В.	учебник	1990	501

Вторая нормальная форма

Таблица находится во 2НФ, если она приведена к 1НФ и каждый неключевой атрибут функционально полно зависит от каждого составного ключа.

Для того чтобы привести таблицу к 2НФ, нужно:

1. построить ее проекцию, исключив атрибуты, которые не находятся в функционально полной зависимости от составного ключа;
2. построить дополнительные проекции на часть составного ключа и атрибуты, функционально зависящие от этой части ключа.

Приведение к 2НФ

Таблица 8.3. Отношение *КНИГИ*, приведённое к 2НФ

<u>ID</u>	<u>Code</u>	<u>Theme</u>	<u>Title</u>	<u>Type</u>	<u>Year</u>	<u>Pg</u>
20	22.18	МК	Язык программирования СИ	учебник	1990	384
10	22.18	МК	Язык АДА	учебник	1988	552
35	32.97	ВТ	Операционные системы ЭВМ	учебное пособие	1992	208
11	32.81	Кибернетика	Общение с ЭВМ на естественном языке	учебник	1982	360
44	32.97	ВТ	ПУ для ПЭВМ	справочник	1992	208
89	32.973	ЭВМ	Интерфейс «человек-компьютер»	учебник	1990	501

Таблица 8.4. Отношение *КНИГИ-АВТОРЫ-РЕДАКТОРЫ* (2НФ)

<u>ID</u>	<u>Author</u>	<u>Editor</u>
20	Бочков С.	Садчиков П.
20	Субботин Д.	Седов П.
10	Джихани Н.	Красилов А.
10	Джихани Н.	Перминов О.
35	Соловьев Г.	
35	Никитин В.	
11	Попов Э.В.	Некрасов А.
44		Витенберг Э.
89	Коутс Р.	Шаньгин В.
89	Влейминк И.	Шаньгин В.

Третья нормальная форма

Третья нормальная форма (3НФ).

Таблица находится в 3НФ, если она находится во 2НФ и в ней отсутствуют функциональные зависимости неключевых атрибутов (атрибутов, не входящих в первичный или потенциальные ключи) от других непервичных атрибутов.

Например, таблица АВТОМОБИЛИ с первичным ключом Государственный номерной знак и атрибутами № кузова и № двигателя, каждый из которых является потенциальным ключом, находится в 3НФ.

Приведение к 3НФ

Таблица 8.5. Отношения КНИГИ, приведённые к 3НФ

<u>ID</u>	<u>Code</u>	<u>Title</u>	<u>Type</u>	<u>Year</u>	<u>Pg</u>
20	22.18	Язык программирования СИ	учебник	1990	384
10	22.18	Язык АДА	учебник	1988	552
35	32.97	Операционные системы ЭВМ	учебное пособие	1992	208
11	32.81	Общение с ЭВМ на естественном языке	учебник	1982	360
44	32.97	ПУ для ПЭВМ	справочник	1992	208
89	32.973	Интерфейс «человек-компьютер»	учебник	1990	501

Табл. 8.6. Отношение РУБРИКАТОР

<u>Code</u>	<u>Theme</u>
22.18	МК
32.97	ВТ
32.973	ЭВМ
32.81	Кибернетика

Отношения, находящиеся в 3НФ, свободны от аномалий модификации.