



**КОМПЬЮТЕРНЫЕ
ТЕХНОЛОГИИ В
ЭКСПЕРТНОЙ
ДЕЯТЕЛЬНОСТИ**

Тема: «Проектирование реляционных баз данных».

Учебные вопросы:

- 1. Нормализация отношений**
- 2. Пример проектирования базы данных**

Литература

основная:

1. **Мишин А.В.** Информационные технологии в профессиональной деятельности: учебное пособие / А.В. Мишин, Л.Е. Мистров, Д.В. Картавцев. – М.: РАП, 2011. – С. 241-259.

дополнительная:

2. **ГОСТ 15971–90.** Системы обработки информации. Термины и определения. – М.: Изд-во стандартов, 1991. – 12 с.
3. **ГОСТ 34.320-96.** Информационные технологии. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы. – М.:ИПК Изд-во стандартов, 2001. – 43 с.

1. Нормализация отношений

Главной целью проектирования БД является получение так называемой «стабильной» логической схемы путем декомпозиции отношений из входной схемы. Концепция декомпозиции очень важна, так как дает возможность вместо целого отношения хранить только его проекции.

Список имен атрибутов A_1, A_2, \dots, A_k отношения R называется **схемой отношения** $R(A_1, A_2, \dots, A_k)$.

Поскольку **порядок следования атрибутов** может быть произвольным и набор возможных отношений заранее не фиксирован, встает задача определения рациональных вариантов группировки атрибутов в отношения (нормализации).

Необходимость нормализации

Основные причины необходимости нормализации отношений:

- уменьшение степени зависимости схем отношений от операций обновления (включения, удаления, модификации атрибутов);
- создание информативной схемы для пользователя БД;
- обеспечение независимости между схемами отношений и изменяемой во времени статистикой запросов.

Требованиям к группировке атрибутов в отношения

Рациональные варианты группировки атрибутов в отношения должны отвечать следующим **требованиям**.

1. Выбранные для отношения **первичные ключи** должны быть **минимальными**.
2. Собранный **состав отношений** базы должен быть **минимальным** (отличаться минимальной избыточностью атрибутов).
3. При выполнении операции включения, удаления и модификации данных в базе не должно быть трудностей.
4. Перестройка набора отношений при введении новых типов данных должна быть минимальной.
5. Разброс времени ответа на различные запросы к БД должен быть небольшим.

Нормальные формы

Степень достижения указанных выше требований определяется нахождением схем отношений в так называемых **нормальных формах**.

Введено шесть нормальных форм (НФ) отношений: 1 НФ, 2 НФ, 3 НФ, НФБК, 4 НФ и 5 НФ.

Характерным для нормальных форм является **распространимость свойств НФ с меньшим номером на НФ с большими номерами**: если отношение находится в 5 НФ, то оно будет соответствовать 4 НФ, 3 НФ и ниже, но не наоборот.

Свойства нормальных форм

5 НФ	Высокая степень независимости между физическим и логическим уровнями			
	Позволяет выявлять новые зависимости, являющиеся логическим следствием исходной схемы. Высокая степень логической независимости			4 НФ
НФБК	Разрешение ситуаций, возникающих в отношениях с перекрывающимися уникальными ключами			
	Удаление транзитивных функциональных зависимостей из контекста отношений		3 НФ	
2 НФ	Учет функциональных зависимостей атрибутов от ключа отношения			
	Позволяет определить язык манипулирования данными и рассматривать БД как совокупность простых таблиц		1 НФ	

Первая нормальная форма

Схема отношения R находится в **1 НФ** только тогда, когда все входящие в нее **атрибуты** являются **атомарными**, т.е. значения соответствующих доменов рассматриваются как неделимые, а не как множества или кортежи из более элементарных доменов:

$$\neg \exists A \subseteq U: A = R_1 \subseteq D_1 \times D_2 \times \dots \times D_k, \\ (D_1, D_2, \dots, D_k) \subset U,$$

где U – полный набор атрибутов.

Вторая нормальная форма

Схема отношения R находится во **2 НФ**, если она находится в 1 НФ и каждый из ее атрибутов полностью зависит от ключа X данного отношения.

Атрибут A полностью зависит от X , если не существует подмножества Z , от которого функционально зависит A , т.е.

$$\neg \exists Z \subseteq X: Z \rightarrow A.$$

Третья нормальная форма

Схема отношения R находится в **3 НФ**, если она находится во 2 НФ и ни один из ее второстепенных атрибутов A_1 транзитивно не зависит ни от одного из возможного ключа X этого отношения.

Для отношения $R(X, Y, Z)$, где $X \cup Y \cup Z = \{A_1, A_2, \dots, A_k\}$, считается, что A_1 транзитивно зависит от X , если

$\exists Y: X \rightarrow Y$ и $Y \not\rightarrow Y$ и $Y \rightarrow A_i$, где $A_i \in X, Y, Z$ ($i = 1, 2, \dots, k$).

Усиленная третья нормальная форма

Схема отношения R находится в **усиленной 3 НФ** (нормальной форме Бойса-Кодда – НФБК), если каждый детерминант является уникальным ключом.

Детерминантом отношения R называется атрибут, от которого полностью зависит некоторый атрибут этого же отношения.

Разница между определениями 3 НФ и НФБК состоит в том, что в определении 3 НФ не предусмотрено разрешение ситуации, возникающей в отношении с перекрывающимися уникальными ключами.

Два ключа $X = \{A_1, A_2, \dots, A_l\}$ и $Y = \{A_m, \dots, A_n\}$ перекрывают друг друга, если $X \cap Y \neq \emptyset$, где $X, Y \subset R$.

Пример отношения «Экзамен»

Атрибуты отношения: **A** (студент), **B** (преподаватель), **C** (предмет).

Семантические ограничения: каждый студент сдает экзамен по определенному предмету только одному преподавателю; каждый преподаватель принимает экзамен только по одному предмету; экзамен по каждому предмету принимается несколькими преподавателями.

Пример таблицы отношения «Экзамен».

Экзамен

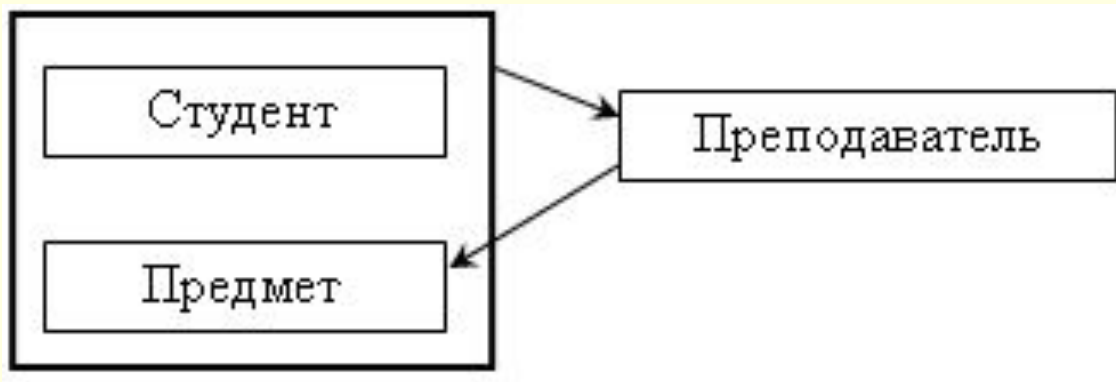
Студент	Преподаватель	Предмет
Глебов	Мишин	Математика и информатика
Дерин	Мишин	Математика и информатика
Шейна	Мистров	Математика и информатика
Башлыкова	Филатов	Судебная статистика
Погодин	Голиков	Судебная статистика

Пример отношения «Экзамен»

Из ограничений имеем, функциональные зависимости:

$$B \rightarrow (A, C), \quad C \rightarrow B, \quad B \neg \rightarrow C.$$

Следовательно, имеет место ситуация:



В этой ситуации перекрываются два возможных ключа (A, C) и (A, B) , т.е. отношение находится в 3 НФ, но не в НФБК. Чтобы избежать трудности, связанные с обновлением, необходима замена исходной схемы отношения проекциями **AB** (A, B) и **BC** (B, C) .

Четвертая нормальная форма

Схема отношения R находится в **4 НФ**, если всякий раз, когда существует многозначная зависимость $X \twoheadrightarrow Y$ (где Y – непустое множество и не является подмножеством X , а XY состоит не из всех атрибутов R), также существует зависимость $X \rightarrow A$ для любого атрибута A в R :

$\forall A \exists X: X \twoheadrightarrow Y$ и $X \rightarrow A$, при $Y \neq \emptyset$, $Y \cup X = \emptyset$, $XY \neq R$.

В отношении $R(X, Y, A)$ существует **многозначная зависимость** Y от X ($X \twoheadrightarrow Y$), если при заданных значениях атрибутов из X существует множество, состоящее из нуля или более взаимосвязанных значений атрибутов из Y , причем множество значений Y не связано со значениями атрибутов ($X \rightarrow A$).

Пятая нормальная форма

Схема отношения R находится в **5 НФ**, если она находится в 4 НФ, а в ее контексте не существует нетривиальной общей нефункциональной зависимости (ОНЗ) и при дальнейшей декомпозиции этой схемы возникает потеря информации.

Рассмотрение правил вывода ОНЗ опустим, т.к. в реальных СУБД разработчики ограничиваются проектированием схем отношений в 3 НФ.

Выводы

Нормализация является процессом создания реляционной схемы БД, где входом является начальная схема, описывающая предметную область, а выходом – другая схема, описывающая то же приложение, но в лучшем виде (с меньшей избыточностью).

Основу нормализации составляет **метод декомпозиции** исходных схем отношений на несколько проекций. При этом должна существовать возможность получения исходных схем композицией этих проекций.

2. Пример проектирования базы данных

Каждый тип объектов предметной области характеризуется набором некоторых характеристик (аспектов описания), называемых элементами данных.

Элемент данных – порция данных, в контексте использования которой отсутствует способ выделения из нее порции, отличной от неё самой.

Пример: студент характеризуется фамилией, именем, отчеством, номером учебной группы, годом рождения и т. п. Если задаются конкретные значения элементов данных, то определяется конкретный объект предметной области.

Состав описания данных

Описание данных должно включать как описание отдельных элементов данных, так и описание типов связей (отношений) между значениями этих элементов.

Рассмотрим два элемента данных – *A* и *B*:

A – элемент данных, от которого направлена связь;

B – элемент данных, к которому направлена связь.

Графически это может быть изображено следующим образом:



Характеристика **типов связей**

Отношение «**один к одному**» (1:1) представляет собой тип связи, когда одно значение элемента данных **A** (от которого направлена связь) определяет одно и только одно значение элемента данных **B** (к которому направлена связь).

Примеры: 1) коды судов и адреса судов;

2) фамилия, имя, отчество судьи и его личный номер.

Отношение «**один ко многим**» (1:M) представляет собой тип связи, когда одно значение элемента данных **A** (от которого направлена связь) определяет несколько значений элемента данных **B** (к которому направлена связь); и каждое значение элемента данных **B** определяется одним значением элемента данных **A**.

Пример: номера квартир дома и список жильцов дома.

Характеристика типов связей

Отношение «**многие к одному**» (**M:1**) – это отношение, обратное отношению **1: M**.

Отношение «**многие ко многим**» (**M:M**) представляет собой тип связи, когда одно значение элемента данных **A** (от которого направлена связь) определяет несколько значений элемента данных **B** (к которому направлена связь); и каждое значение элемента данных **B** может определяться несколькими значениями элемента данных **A**.

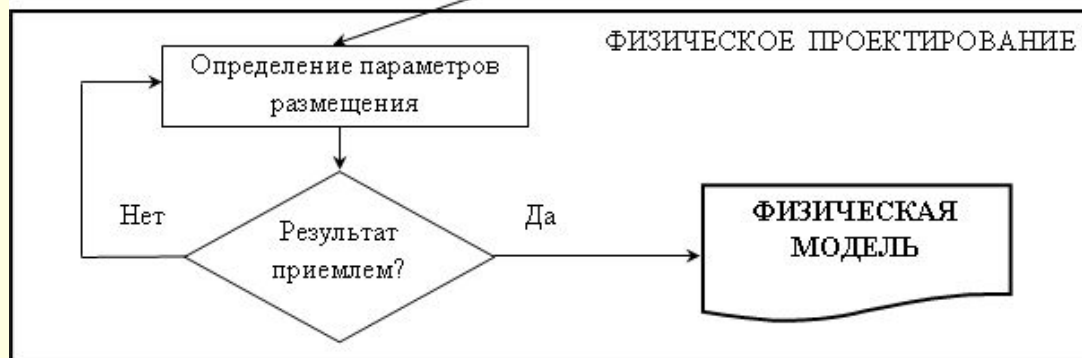
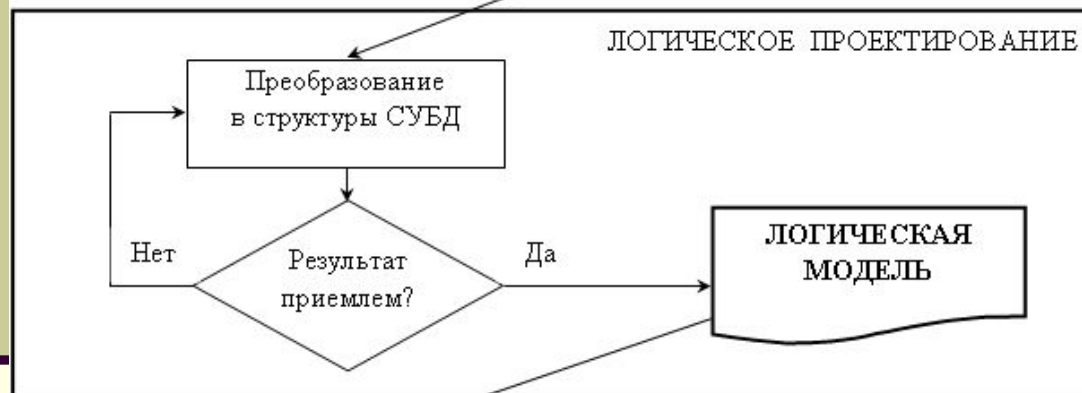
Пример: отношения знакомства между людьми.

Сложности проектирования БД

Основные трудности при проектировании возникают из-за:

- **противоречивости требований к БД** (например, с одной стороны, требуется сокращение объёма БД, а с другой стороны, ускорение поиска информации, хотя, как правило, одно исключает другое);
- **сложности** получения правильного **описания** данных и взаимосвязей между данными;
 - из-за большого **объёма вычислений** и трудоёмкости используемых математических методов.

Этапы проектирования базы данных



Пример **концептуального** проектирования

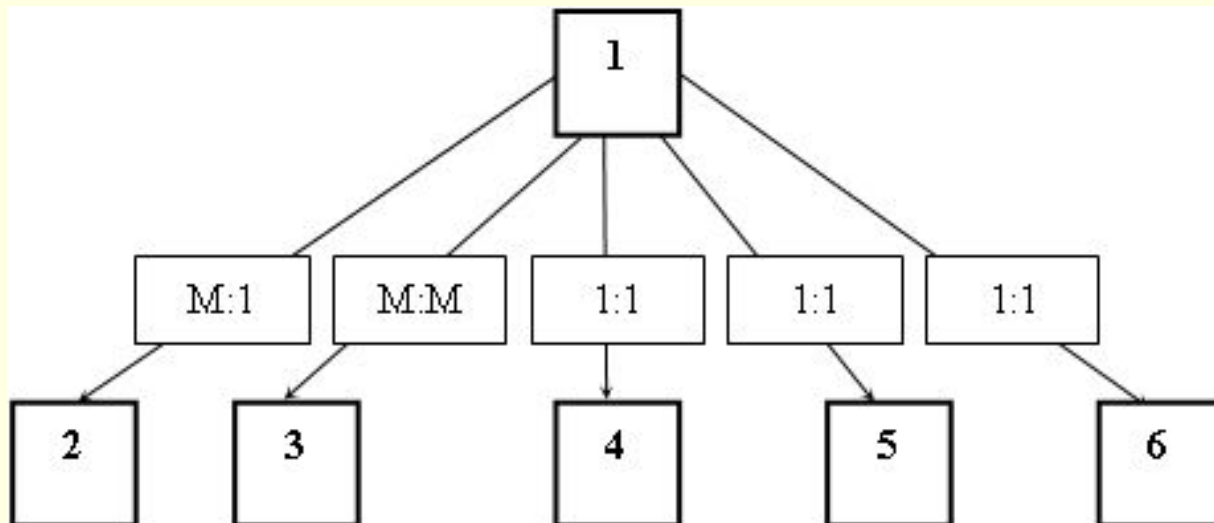
Предположим, что необходимо спроектировать **БД** для информационной системы «**Преступные группировки**», которая должна обеспечивать автоматизацию следующих пользовательских функций:

- определение наиболее опасных преступных группировок;
- определение наиболее опасных членов преступных группировок;
- поиск информации о возможном месте нахождения членов группировок.

Исходные данные

Будем считать, что пользователь на основе какого-либо формального или неформального метода определяет **опасность группировки** на основе следующих **данных**:

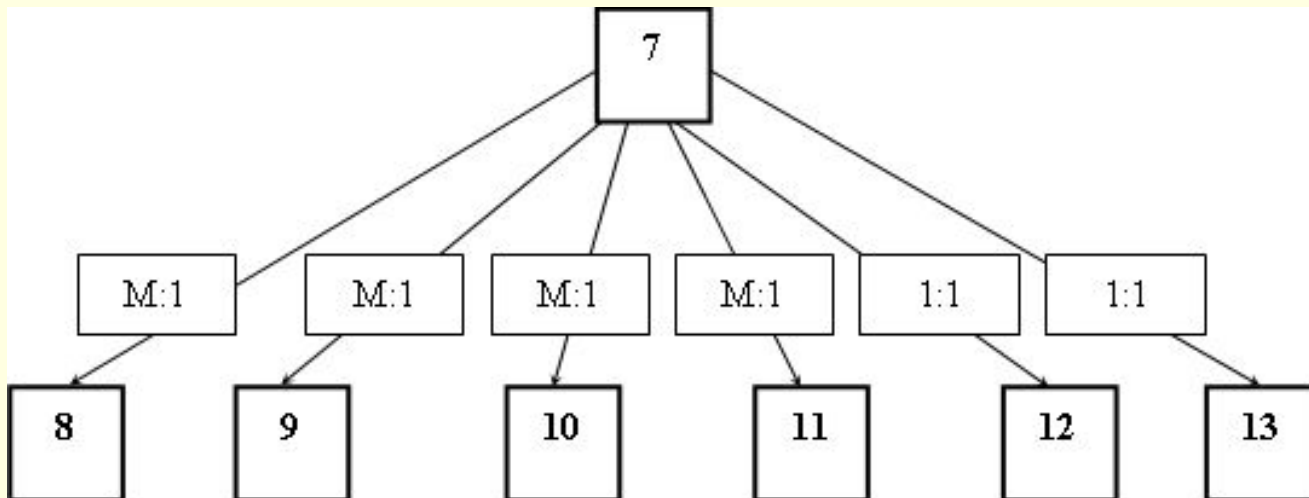
- 1 – условный номер группировки;
- 2 – район деятельности группировки;
- 3 – сферы деятельности группировки;
- 4 – активность группировки;
- 5 – продолжительность деятельности группировки;
- 6 – число членов группировки.



Исходные данные

Степень **опасности члена группировки** определяется по следующим данным:

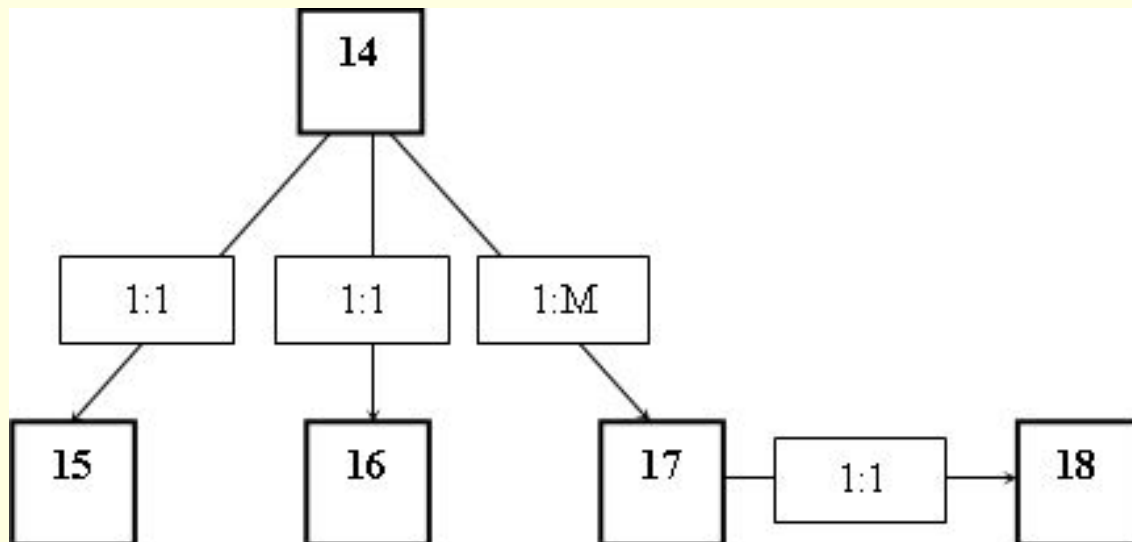
- 7 – фамилия, имя, отчество;
- 8 – группировка, в которую он входит;
- 9 – опасность этой группировки;
- 10 – сфера деятельности группировки;
- 11 – роль в группировке (лидер, скупщик краденного и т. д.);
- 12 – степень активности преступной деятельности;
- 13 – наличие судимости.



Исходные данные

Для поиска **места нахождения членов** преступных группировок будем весьма упрощенно предполагать необходимость следующих данных:

- 14 – фамилия, имя, отчество;
- 15 – адрес;
- 16 – особые приметы;
- 17 – фамилии, имена, отчества знакомых;
- 18 – адреса знакомых.



Переход к **глобальному** представлению

Мы осуществили первый шаг концептуального проектирования – сбор и анализ требований к данным.

Полученное описание данных для каждой пользовательской функции называется **внешним представлением данных**.

Следующий шаг – редактирование – подразделяется на два этапа:

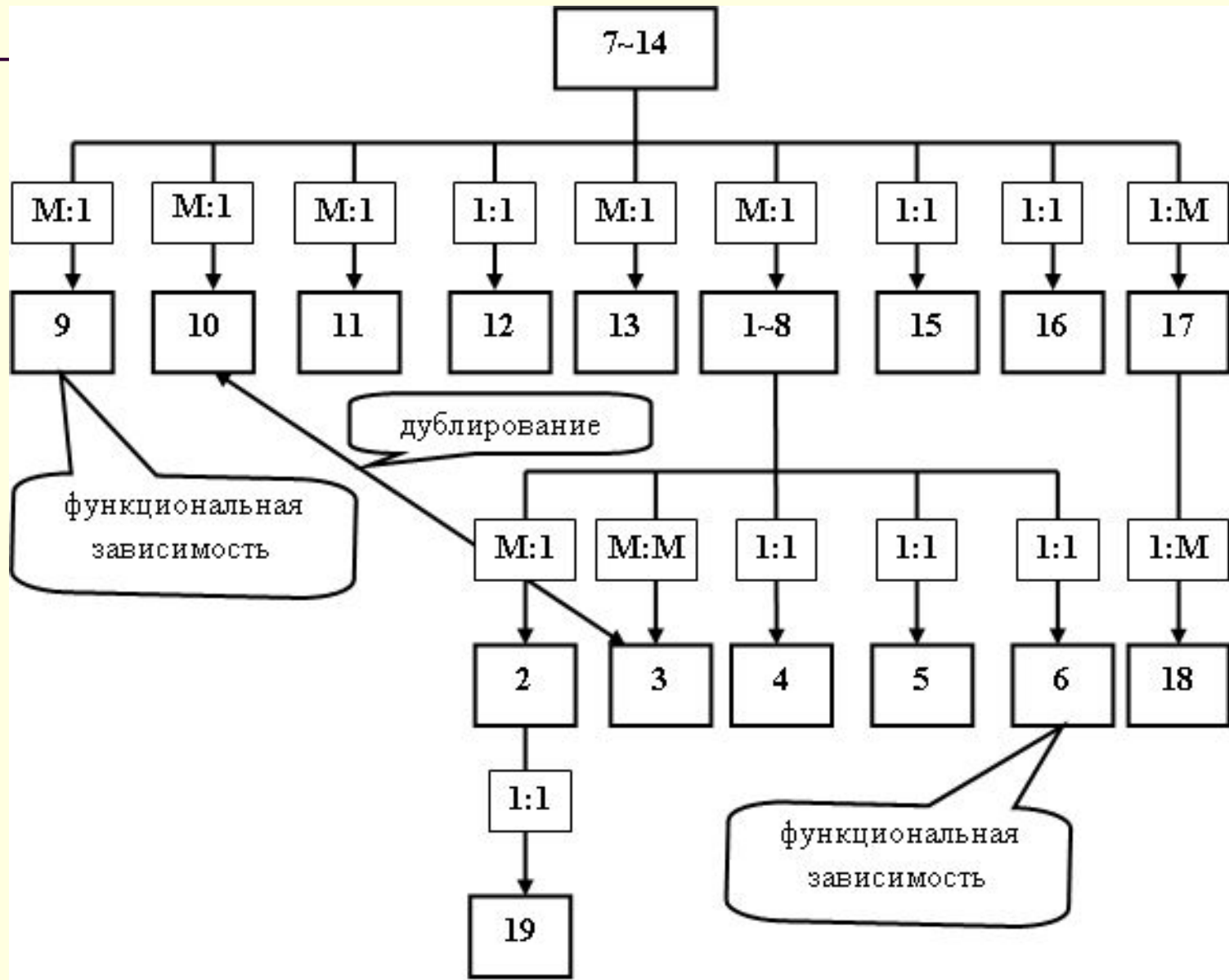
- 1) получение глобального представления данных;
- 2) получение внутреннего представления данных.

Глобальное представление данных – это представление, содержащее полную совокупность требований к данным.

Переход к **глобальному** представлению

Пользовательские функции, входящие во внешнее представление, могут задаваться разными пользователями, использующими различную терминологию. Поэтому элементы данных в них могут частично противоречить друг другу или дублироваться. Необходима идентификация элементов данных, входящих в разные пользовательские функции, и устранение противоречий в их описании. Например, **8** – «**группировка**» и **1** – «**наименование группировки**» очевидно одно и то же. То же можно сказать и о **7** и **14** – «**фамилия, имя, отчество**». **5** – «**продолжительность деятельности группировки**» лучше заменить «**датой начала деятельности**», чтобы этот элемент данных не зависел от времени заполнения БД. **2** – «**район деятельности**» целесообразно дополнить элементом **19** – «**код района**» для связи с уже существующими БД.

Глобальное представление данных



Переход к внутреннему представлению

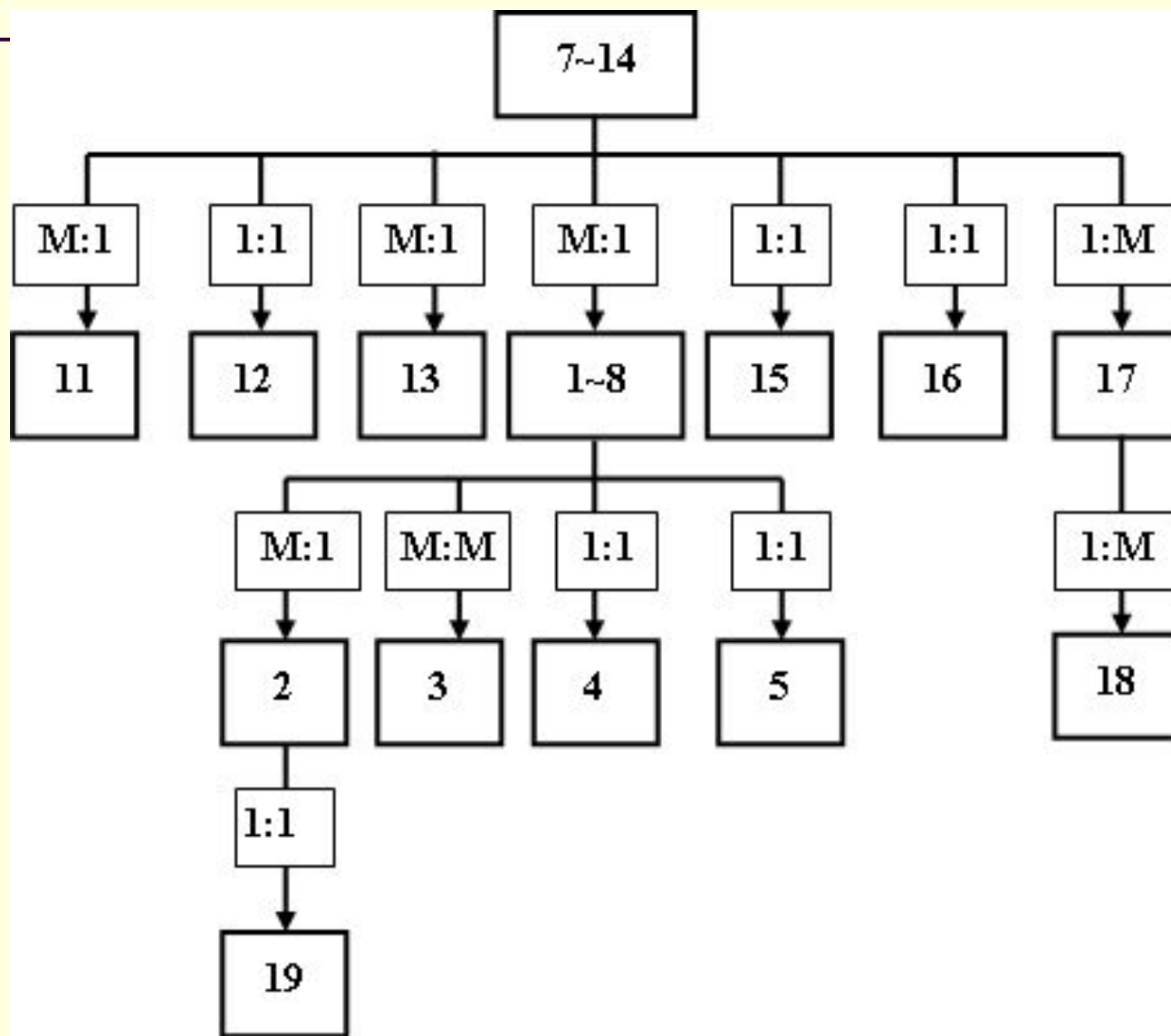
Внутреннее представление данных – это те и только те данные, которые необходимы для автоматизации пользовательских функций, т.е. это некоторая часть глобального представления, в которой устранена избыточность элементов данных.

Избыточность могла возникнуть либо из-за дублирования элементов данных, либо по причине функциональной зависимости данных.

В нашем примере:

- дважды присутствует, т.е. дублируется, элемент данных «Сфера деятельности группировки» – **10** и **3**;
- элемент данных **6** – «Число участников группировки» легко вычисляется по имеющимся данным;
- то же относится и к элементу данных **9** – «Опасность группировки».

Внутреннее представление данных



Логическое проектирование

Таблица 1*

Шаг 1. Приведение к 1 НФ:

- 1) все элементы данных атомарные, т.е. представляют собой одиночные значения, а не их объединение;
- 2) отсутствуют повторяющиеся группы элементов данных, т.е. элементы данных, имеющие одинаковое смысловое значение, объединяются. В полученной выше концептуальной модели базы данных неатомарным является элемент данных «*Фамилия, имя, отчество*». Он может быть разделен на атомарные так, чтобы не появились повторяющиеся группы полей. **Ключевые элементы данных будем выделять серым цветом в таблицах и полужирным шрифтом в тексте.**

7-1	Фамилия
7-2	Имя
7-3	Отчество
11	Роль в группировке
12	Степень активности преступной деятельности
13	Наличие судимости
15	Адрес
16	Особые приметы
8	Группировка, в которую он входит
2	Район деятельности группировки
19	Код района
3	Сфера деятельности группировки
4	Активность группировки
5	Дата начала деятельности группировки
17	Фамилия, имя, отчество знакомого
18	Адрес знакомого

Логическое проектирование

Шаг 2. Для приведения к 2 НФ необходимо в каждой таблице:

- 1) определить **ключевые** элементы данных;
- 2) определить **функциональные** зависимости **неключевых** элементов данных **от ключевых** элементов;
- 3) **разделить** элементы данных **на группы** по функциональным зависимостям так, чтобы каждый неключевой элемент данных определялся полным набором ключевых элементов;
- 4) определить **связанные** между собой **группы** элементов данных и типы отношений между ними;
- 5) для реализации связей между группами элементов данных **дополнить** их элементами данных, организующих связь.

Логическое проектирование

При построении **2 НФ** следует помнить следующие **правила**:

- 1)** если ключ является простым (состоящим из одного элемента данных), то требование 2 НФ для этой группы элементов автоматически выполняется;
- 2)** если между группами элементов **A** и **B** существует связь типа **1:M**, то для реализации этой связи группа элементов **B** дополняется ключевыми элементами данных группы элементов **A**;
- 3)** если в группе элементов данных возможен альтернативный выбор элементов данных для реализации связи, то следует выбирать элементы меньшей суммарной длины.

Приведение к 2 НФ

Выполним описанные выше действия. Анализ функциональных зависимостей показывает, что могут быть выделены **четыре группы** функциональных зависимостей:

$$(8) \leftarrow (2, 19, 5), \quad (A)$$

$$(8, 3) \leftarrow (4), \quad (B)$$

$$(7-1, 7-2, 7-3) \leftarrow (11, 12, 13, 15, 16), \quad (C)$$

$$(17) \leftarrow (18), \quad (D)$$

где стрелка направлена от неключевых элементов группы к ключевым элементам.

Между данными группами можно выделить следующие отношения:

$$A \rightarrow B \text{ типа } (1:M),$$

$$A \rightarrow C \text{ типа } (1:M),$$

$$D \rightarrow C \text{ типа } (M:M).$$

Приведение к 2 НФ

Выполним описанные выше действия. Анализ функциональных зависимостей показывает, что могут быть выделены **четыре группы** функциональных зависимостей:

$$(8) \leftarrow (2, 19, 5), \quad (A)$$

$$(8, 3) \leftarrow (4), \quad (B)$$

$$(7-1, 7-2, 7-3) \leftarrow (11, 12, 13, 15, 16), \quad (C)$$

$$(17) \leftarrow (18), \quad (D)$$

где стрелка направлена от неключевых элементов группы к ключевым элементам.

Приведение к 2 НФ

Между группами можно выделить следующие **отношения**:

$A \rightarrow B$ типа (1:M),

$A \rightarrow C$ типа (1:M),

$D \rightarrow C$ типа (M:M).

В этом случае для организации связи между группами элементов по существующему правилу следует:

- включить ключевые поля группы **A** в группу **B**, что фактически уже осуществлено при определении функциональных зависимостей;
- включить ключевое поле группы **A** в группу **C**;
- создать так называемую «**таблицу развязки**» между группами **D** и **C**, которая составлена из ключевых элементов как группы **D**, так и группы **C**.

Таблица 2*

1:M	8	Группировка, в которую он входит
	2	Район деятельности группировки
	19	Код района
	5	Дата начала деятельности группировки

Таблица 3*

8	Группировка, в которую он входит	
	3	Сфера деятельности группировки
	4	Активность группировки

Таблица 4*

1:M	7-1	Фамилия	
		7-2	Имя
		7-3	Отчество
	11	Роль в группировке	
	12	Степень активности преступной деятельности	
	13	Наличие судимости	
	15	Адрес	
	16	Особые приметы	
M:M	8	Группировка, в которую он входит	

Таблица 5*

7-1	Фамилия	
	7-2	Имя
	7-3	Отчество
17	Фамилия, имя, отчество знакомого	

Таблица 6*

1:M	17	Фамилия, имя, отчество знакомого
	18	Адрес знакомого

Таблицы в 2 НФ

Приведение к 3 НФ

Шаг 3. Для приведения к **3 НФ** таблиц, уже приведенных к 2 НФ, необходимо добиться, чтобы ни один неключевой элемент в этих таблицах не определялся другим неключевым элементом, т.е. должны отсутствовать транзитивные зависимости.

Анализ таблиц 2^*-6^* показывает, что указанному требованию **не удовлетворяет только таблица 2^*** . Действительно,

$$(8) \leftarrow (2) \leftarrow (19),$$

$$(8) \leftarrow (19) \leftarrow (2).$$

Устранение этих зависимостей может быть осуществлено разбиением на следующие группы элементов данных:

$$(8) \leftarrow (19, 5), \quad (E)$$

$$(19) \leftarrow (2), \quad (F)$$

Для связи между группами элементов E и F использован элемент **19**, как имеющий меньшую длину.



Таблицы в 3 НФ

Подготоввил

А.В. Мишин

31.01.2016

Спасибо за



внимание!

**Кафедра права и информатики,
информационного права и
естественно-научных дисциплин
Центрального филиала
Российского государственного
университета правосудия**