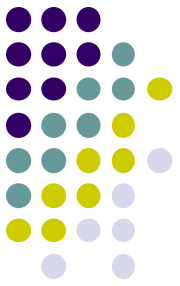


# Базы данных

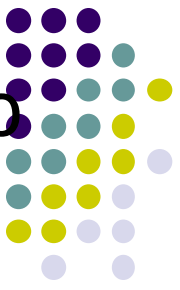
Раздел №1. Основы построения баз данных

Тема 3. Проектирование реляционных баз  
данных



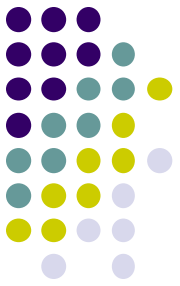
# Учебные вопросы.

1. Проектирование реляционных БД с использованием нормализации отношений
2. Проектирование реляционных БД на основе концептуальной модели



- Выделяют два подхода к проектированию реляционной базы данных.
- **Первый** подход был предложен Коддом и заключается в создании на этапе концептуального проектирования не концептуальной модели данных, а непосредственно **реляционной схемы базы данных**.
- **Второй** подход основан на **концептуальной модели данных**, создаваемой на этапе концептуального проектирования.

# 1. Проектирование реляционных БД с использованием нормализации отношений



Исходной информацией является представление предметной области в виде одного или нескольких отношений.

Одни и те же данные могут группироваться в таблицы-отношения различными способами.

# Фундаментальные свойства отношений

- 1. Отсутствие кортежей-дубликатов.** Исходя из этого свойства, у каждого отношения должен быть первичный ключ.
- 2. Отсутствие упорядоченности кортежей.** Дает дополнительную гибкость СУБД при хранении баз данных во внешней памяти и при выполнении запросов к базе данных.
- 3. Отсутствие упорядоченности атрибутов.** Для ссылки на значение атрибута в кортеже отношения всегда используется имя атрибута.
- 4. Атомарность значений атрибутов.** Это следует из определения домена как множества значений простого типа данных.

# Реляционная модель данных.

## Общие понятия

**Составной ключ** —

это ключ, содержащий более чем один атрибут.

**Простой ключ** —

это ключ, состоящий из одного атрибута.

Ключи обеспечивают **основной механизм адресации на уровне кортежей в реляционной системе.**

**Внешним ключом** —

атрибут или группа атрибутов одного отношения, которым назначена ссылка на первичный ключ другого отношения, для однозначного его определения.

# Типы отношений между сущностями

1. Связь «один к одному» (**1:1**) — такой тип связи между типами сущностей *A* и *B*, когда каждому экземпляру сущности *A* соответствует один и только один экземпляр сущности *B* и наоборот.
2. Связь «один ко многим» (**1:M**) — одному экземпляру сущности *A* может соответствовать 0, 1 или несколько экземпляров сущности *B*, однако каждому экземпляру сущности *B* соответствует только один экземпляр сущности *A*.
3. Связь «многие к одному» (**M:1**) — является обратной связи 1:M.
4. Связь «многие ко многим» (**M:M**) — каждому экземпляру сущности *A* может соответствовать 0, 1 или несколько экземпляров сущности *B* и наоборот.

# Проектирование реляционных БД

- В базе данных отражается информация об определенной предметной области.
- Предметной областью называется часть реального мира, представляющая интерес для данного исследования (использования).
- Вся предметная область может быть представлена в виде одного универсального отношения.
- Недостатки такого способа хранения устраняются нормализацией.
- Для реляционной базы данных проектирование логической структуры заключается в том, чтобы разбить всю информацию по таблицам (отношениям), а также определить состав полей (атрибутов) для каждой из этих таблиц.



# Универсальное отношение

<b>ФИО студента</b>	<b>Факультет</b>	<b>Специальность</b>	<b>Дисциплина</b>	<b>Вид</b>	<b>Оценка</b>	<b>Преподаватель</b>
Андреев А. А.	ПММ	Математика	СУБД	Экз	3	Борисов Б. Б.
Андреев А. А.	ПММ	Математика	Ин. Яз	Экз	4	Васильев В. В.
Андреев А. А.	ПММ	Математика	БД	Экз	5	Гаврилов Г. Г.
Андреев А. А.	ПММ	Математика	ФП	Зач	3	Дмитриев Д. Д.
Андреев А. А.	ПММ	Математика	Физика	Экз	3	Егоров Е. Е.
Андреев А. А.	ПММ	Математика	СУБД	КР	4	Петров П. П.
Андреев А. А.	ПММ	Математика	ВССТ	Зач	2	Сидоров С. С.
Денисов Д. Д.	ПММ	Механика	СУБД	Экз	3	Борисов Б. Б.
Денисов Д. Д.	ПММ	Механика	Ин. Яз	Экз	3	Васильев В. В.
Денисов Д. Д.	ПММ	Механика	БД	Экз	3	Гаврилов Г. Г.
Денисов Д. Д.	ПММ	Механика	ФП	Зач	2	Дмитриев Д. Д.
Денисов Д. Д.	ПММ	Механика	Физика	Экз	3	Егоров Е. Е.
Денисов Д. Д.	ПММ	Механика	СУБД	КР	2	Петров П. П.
Денисов Д. Д.	ПММ	Механика	ВССТ	Зач	2	Сидоров С. С.

# Недостатки универсального отношения

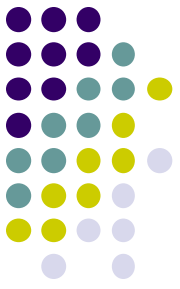
- **Дублирование информации (избыточность).** У студентов, обучающихся на одном факультете, будет повторяться название факультета. Для разных факультетов будут повторяться специальности.
- **Потенциальная противоречивость (аномалии обновления).** Если, например, изменится название специальности, то изменяя её в одном кортеже (у одного студента), необходимо изменять и во всех других кортежах, где она присутствует.
- **Потенциальная возможность потери сведений (аномалии удаления).** При удалении информации о всех студентах, поступающих на определенную специальность, мы теряем все сведения об этой специальности.
- **Потенциальная возможность невключения информации в базу данных (аномалии включения).** В базу данных сложно внести сведения о специальности, если на ней нет обучающихся студентов.



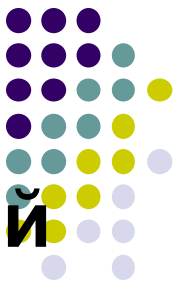
- Определенный набор отношений обладает лучшими свойствами при включении, модификации и удалении данных, если он отвечает определенным требованиям **нормализации отношений**.
- **Нормализация отношений** - это формальный аппарат ограничений на их формирование, который позволяет устранить дублирование данных, обеспечить их непротиворечивость и уменьшить затраты на поддержание БД.

Группировка атрибутов в отношении должна отвечать следующим требованиям:

- выбранные для отношений первичные ключи должны быть минимальными;
- выбранный состав отношений базы должен быть минимальным (отличаться минимальной избыточностью атрибутов);
- при выполнении операций включения, удаления и модификации данных в базе не должно быть трудностей (аномалий);
- перестройка набора отношений при введении новых типов данных должна быть минимальной;



- Процесс проектирования представляет собой *процесс нормализации схем отношений*, причем каждая следующая нормальная форма обладает лучшими свойствами, чем предыдущая.
- Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет соответствующему набору ограничений.



- Методы нормализации базируются на использовании понятий **функциональной и транзитивной зависимости**.
- *Функциональная зависимость*. В отношении  $R$  атрибут  $Y$  функционально зависит от атрибута  $X$  ( $X$  и  $Y$  могут быть составными) в том и только в том случае, если каждому значению  $X$  соответствует в точности одно значение  $Y$ , т.е.  $R.X \rightarrow R.Y$ .

# Функциональные зависимости (зависимости между атрибутами отношения)

Пусть  $R(A_1, A_2, \dots, A_n)$  – схема отношения,  
а  $X$  и  $Y$  – подмножества  $\{A_1, A_2, \dots, A_n\}$ .

Функциональная зависимость на отношении  $R$  – это утверждение вида:

**"Если два кортежа  $R$  совпадают по атрибутам множества  $X$  (т. е. эти кортежи имеют в соответствующих друг другу компонентах одни и те же значения для каждого атрибута множества  $X$ ), то они должны совпадать и по атрибутам множества  $Y$  ."**

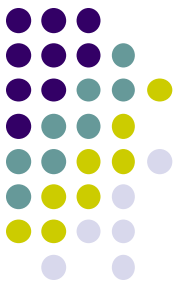
Формально эта зависимость записывается выражением  $X \rightarrow Y$ , причем говорится, что  $X$  функционально определяет  $Y$ .

Часто используется другое утверждение:  $X$  функционально определяет  $Y$  или  $Y$  функционально зависит от  $X$  тогда и только тогда, когда каждое значение множества  $X$  отношения  $R$  связано с одним значением множества  $Y$  отношения  $R$ .  
Иначе говоря, если два кортежа  $R$  совпадают по значению  $X$ , они совпадают и по значению  $Y$ .



Полной функциональной зависимостью называется зависимость неключевого атрибута от всего составного ключа.

Частичной функциональной зависимостью будем называть зависимость неключевого атрибута от части составного ключа.



- *Транзитивная функциональная зависимость.* Функциональная зависимость  $R.X \rightarrow R.Y$  называется транзитивной, если существует такой атрибут  $Z$ , что имеются функциональные зависимости  $R.X \rightarrow R.Z$  и  $R.Z \rightarrow R.Y$  и отсутствует функциональная зависимость  $R.Z \rightarrow R.X$ .

Для решения перечисленных  
задач выполняется  
нормализация исходных схем  
отношений проекта БД.

# Последовательность нормальных форм реляционных баз данных:

- первая нормальная форма (**1NF**);
- вторая нормальная форма (**2NF**);
- третья нормальная форма (**3NF**);
- нормальная форма Бойса—Кодда (**BCNF**);
- четвертая нормальная форма (**4NF**);
- пятая нормальная форма или нормальная форма проекции-соединения (**5NF** или **PJ/NF**).

Все отношения (нормализованные и ненормализованные)

Отношения в 1НФ (нормализованные отношения)

Отношения в 2НФ

Отношения в 3НФ

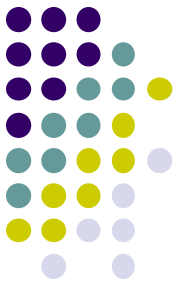
Отношения в 4НФ

Отношения в 5НФ

**ОДИН ФАКТ ХРАНИТСЯ ОДИН РАЗ!!!**

# Свойства нормальных форм:

- каждая следующая нормальная форма, в некотором смысле, лучше предыдущей нормальной формы;
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются;
- Нормализация отношений заключается в декомпозиции отношения, находящегося в предыдущей нормальной форме, в два или более отношения, удовлетворяющих требованиям следующей нормальной формы.

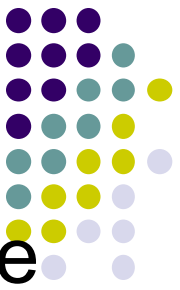


- Для реляционных БД необходимо, чтобы все отношения базы данных находились обязательно в 1NF.
- Нормальные формы более высокого уровня могут использоваться разработчиками по своему усмотрению.

- Отношение называется нормализованным или приведенным к первой нормальной форме (1NF), если все его атрибуты простые или атомарные (неделимые).

Отношение, находящееся в первой нормальной форме, имеет следующие свойства:

- в отношении нет одинаковых кортежей;
- кортежи не упорядочены;
- атрибуты не упорядочены и различаются по наименованиям;
- все значения атрибутов атомарные.

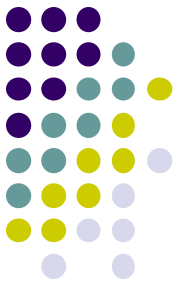




## Пример. Информационная система «Преступные группировки».

Информация о членах преступных группировок:

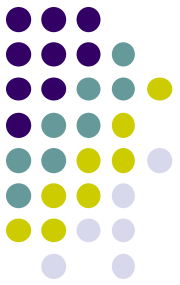
- фамилия, имя, отчество;
- роль в группировке (лидер, исполнитель, скупщик краденного и т.п.);
- предыдущие судимости: статьи УК, сроки лишения свободы, даты освобождения;
- адрес проживания;
- особые приметы.





## Информация о группировках:

- условный номер группировки;
- район деятельности;
- условный номер района (для связи с другими БД);
- сфера деятельности;
- активность деятельности в данной сфере;
- зафиксированная дата начала деятельности.



- В описании информационной системы неатомарными являются атрибуты «Фамилия, имя, отчество» и «Предыдущие судимости».
- Они могут быть разделены на атомарные таким образом, чтобы не появились повторяющиеся группы полей.



№	Элементы данных	Ключевые элементы данных
1	2	3
1	Фамилия члена преступной группировки	К
2	Имя члена преступной группировки	К
3	Отчество члена преступной группировки	К
4	Роль в группировке	
5	Предыдущая судимость: статья УК	
6	Предыдущая судимость: срок лишения свободы	
7	Предыдущая судимость: дата освобождения	К
8	Адрес	
9	Особые приметы	
10	Условный номер группировки	К
11	Район деятельности	
12	Условный номер района	К
13	Сфера деятельности	
14	Активность деятельности в данной сфере	
15	Зафиксированная дата начала деятельности	

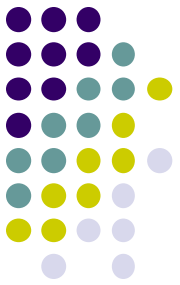
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

База данных, приведенная к 1NF

Для приведения отношения к **2NF**  
необходимо:

- определить ключевые атрибуты;
- определить функциональные зависимости неключевых атрибутов от ключевых;
- разделить атрибуты на группы по функциональным зависимостям так, чтобы каждый неключевой атрибут определялся полным набором ключевых атрибутов;

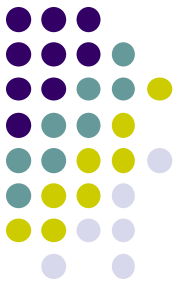


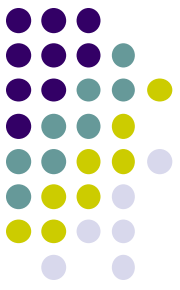


- определить связанные между собой группы атрибутов и типы отношений между ними;
- для реализации связей между группами атрибутов дополнить их атрибутами, организующих связь.

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

- если ключ является простым, то требование **2NF** для отношения автоматически выполняется;
- если между отношениями A и B существует связь типа 1:M, то при организации связи отношение B дополняется ключевыми атрибутами отношения A;
- если в отношении возможен альтернативный выбор атрибутов для организации связи, то следует выбирать атрибуты меньшей длины.





- В примере могут быть выделены четыре группы функциональных зависимостей:

$$(1, 2, 3, 7) \leftarrow (5, 6) \quad (A)$$

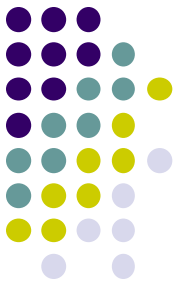
$$(1, 2, 3) \leftarrow (4, 8, 9) \quad (B)$$

$$(10, 13) \leftarrow (14) \quad (C)$$

$$(10) \leftarrow (11, 12, 15) \quad (D)$$

- Стрелка направлена от неключевых элементов группы к ключевым.





- Между данными группами можно выделить следующие отношения:

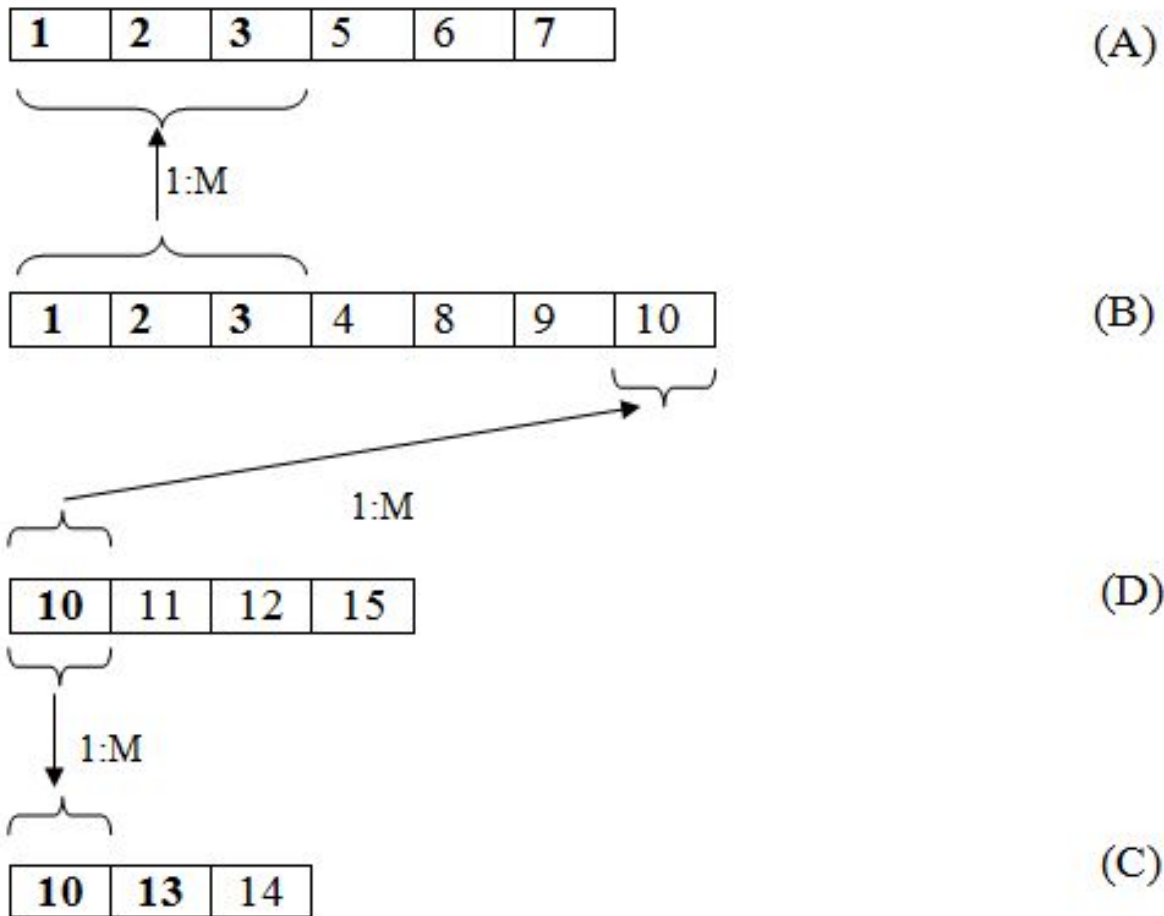
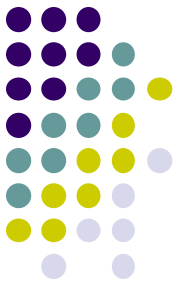
$B \rightarrow A$  типа 1:M,

$D \rightarrow B$  типа 1:M,

$D \rightarrow C$  типа 1:M.

В этом случае для организации связи между атрибутами необходимо:

- включить ключевые поля группы B в группу A;
- включить ключевое поле группы D в группы B и C.



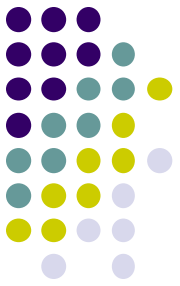
- База данных, приведенная к 2NF



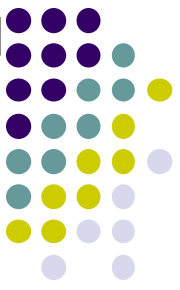
Для приведения к **3NF** необходимо в каждом отношении

- определить транзитивные зависимости между атрибутами;
- разделить атрибуты на группы по транзитивным зависимостям так, чтобы ни один неключевой атрибут в этих группах не определялся другим неключевым атрибутом;

- определить связанные между собой группы атрибутов и типы отношений между ними;
- для реализации связей между группами атрибутов дополнить их атрибутами, организующими связь.



Организация связей осуществляется таким же образом, как и при построении **2NF**.



- Транзитивные зависимости имеются только в отношении D:

$$(10) \leftarrow (11) \leftarrow (12)$$

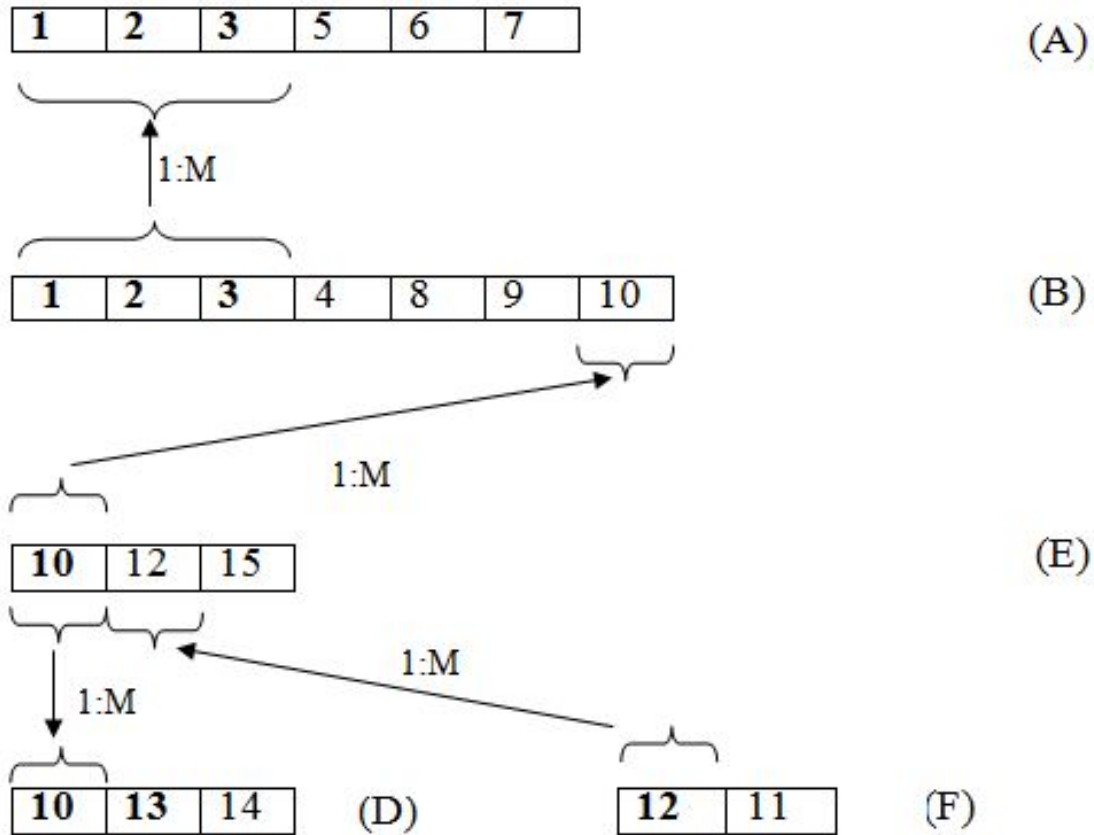
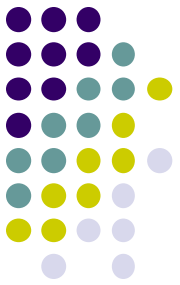
$$(10) \leftarrow (12) \leftarrow (11)$$

- Устранение этих зависимостей может быть осуществлено разбиением на следующие группы атрибутов:

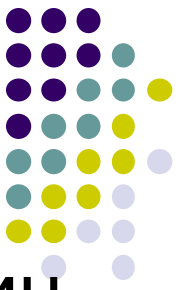
$$(10) \leftarrow (12, 15) \quad (E)$$

$$(12) \leftarrow (11) \quad (F)$$

- Для связи между группами элементов E и F использован элемент 12, как имеющий меньшую длину.



- База данных, приведенная к 3NF

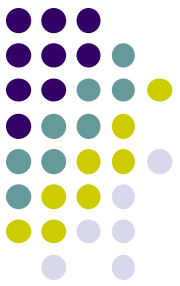


- *Нормальная форма Бойса-Кодда BCNF* требует, чтобы в таблице был только один потенциальный первичный ключ.
- Если обнаружился второй столбец, позволяющий однозначно идентифицировать строку, то для приведения к нормальной форме Бойса-Кодда такие данные надо вынести в отдельную таблицу.



- Для приведения отношения к **четвертой нормальной форме** необходимо устранить имеющиеся в нем многозначные зависимости.
- Нужно, чтобы вставка или удаление любой строки таблицы не требовали бы вставки, удаления или модификации других строк этой же таблицы.





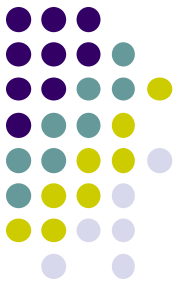
- Отношение, находящееся в четвертой нормальной форме, в некоторых случаях еще можно разбить на три или более таблицы, при соединении которых получается исходная таблица.
- В результате такой, как правило, искусственной декомпозиции образуется таблица, находящаяся в *пятой нормальной форме*.

Итого:

- Рассмотренный декомпозиционный подход к проектированию схемы реляционной БД путем последовательной нормализации первоначальных отношений используется при условии небольшого числа атрибутов.
- В случаях когда число атрибутов превышает 20, этот метод становится излишне громоздким.



## 2. Проектирование реляционных БД на основе концептуальной модели

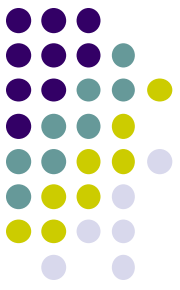


### 1. Инфологическое моделирование

- Проектирование БД начинается с предварительной структуризации предметной области: объекты реального мира подвергаются классификации, фиксируется совокупность подлежащих отображению в БД типов объектов.



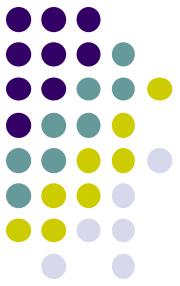
- Для каждого типа объектов фиксируется совокупность свойств, посредством которых будут описываться конкретные объекты этого типа в БД, виды отношений (взаимосвязей) между этими объектами.
- Затем решаются вопросы о том, какая информация об этих объектах должна быть представлена в БД, как ее представить с помощью данных.



- **Инфологическая модель** предметной области (ИЛМ) — это описание предметной области, выполненное без ориентации на используемые в дальнейшем программные и технические средства.
- Описание инфологической модели называется **схемой** (**концептуальной схемой**).

# Основные требования к ИЛМ:

- адекватное отображение предметной области;
- непротиворечивость;
- отсутствие неоднозначности трактовки;
- возможность легкой расширяемости, обеспечивающая ввод данных без изменения ранее определенных;
- обеспечение возможности композиции и декомпозиции модели.





- **На этом этапе используется модель «сущность-связь».**
- Основное ее назначение — семантическое описание предметной области и представление информации для обоснования выбора видов моделей и структур данных, которые в дальнейшем будут использованы в системе.



- При моделировании предметной области проектировщик разбивает ее на ряд локальных областей, моделирует каждое локальное представление, а затем их объединяет.
- Чаще всего локальная модель отражает представление отдельного пользователя. Иногда такая модель может описывать и некоторую независимую область данных нескольких функциональных задач.

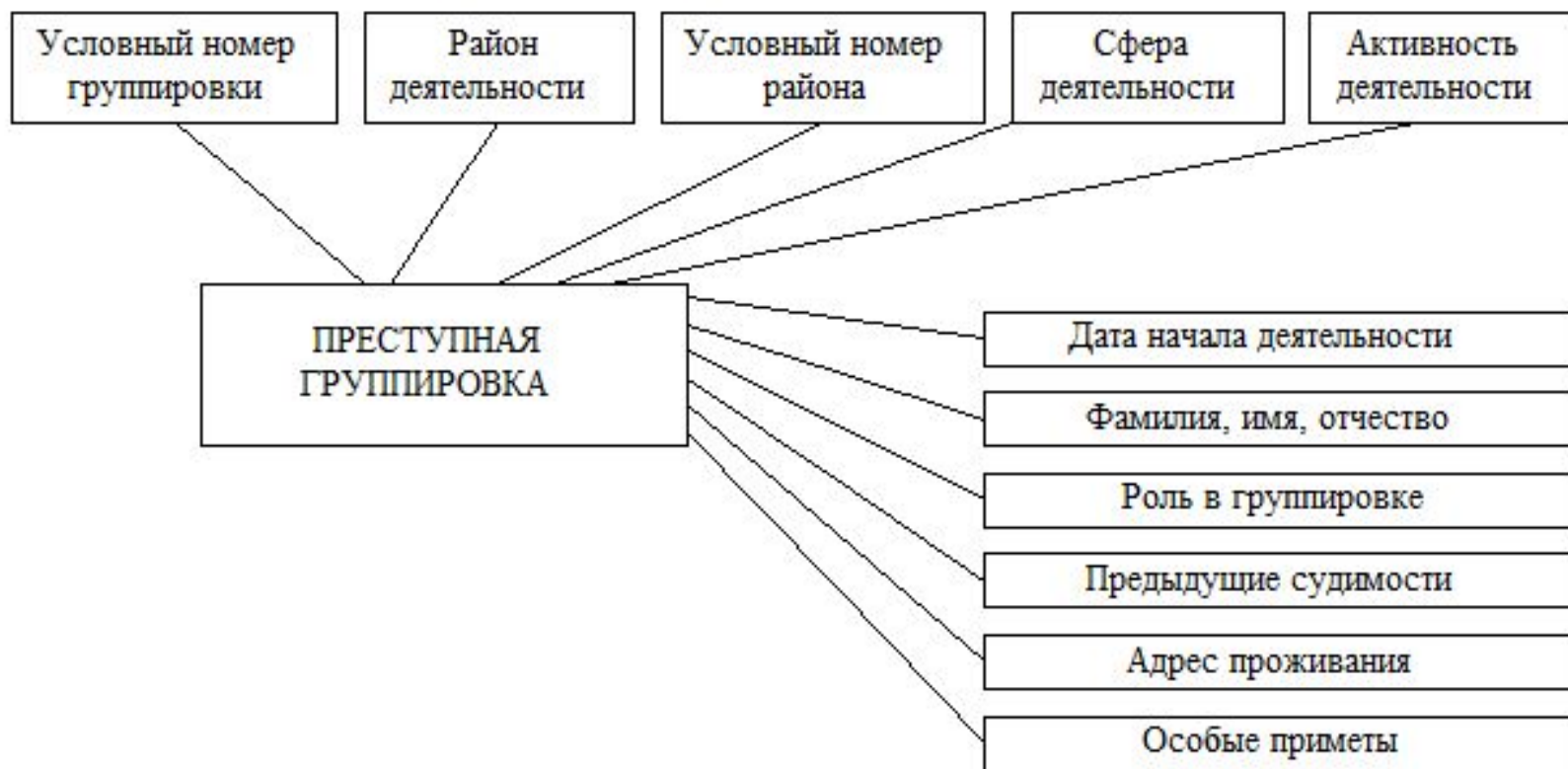




- Для удобства проектирования в отдельном локальном представлении желательно использовать шесть-семь типов сущностей.
- Для каждого локального представления необходимо сформулировать сущности, требуемые для его описания, т.е. указать типы объектов предметной области, о которых в системе должна накапливаться информация.

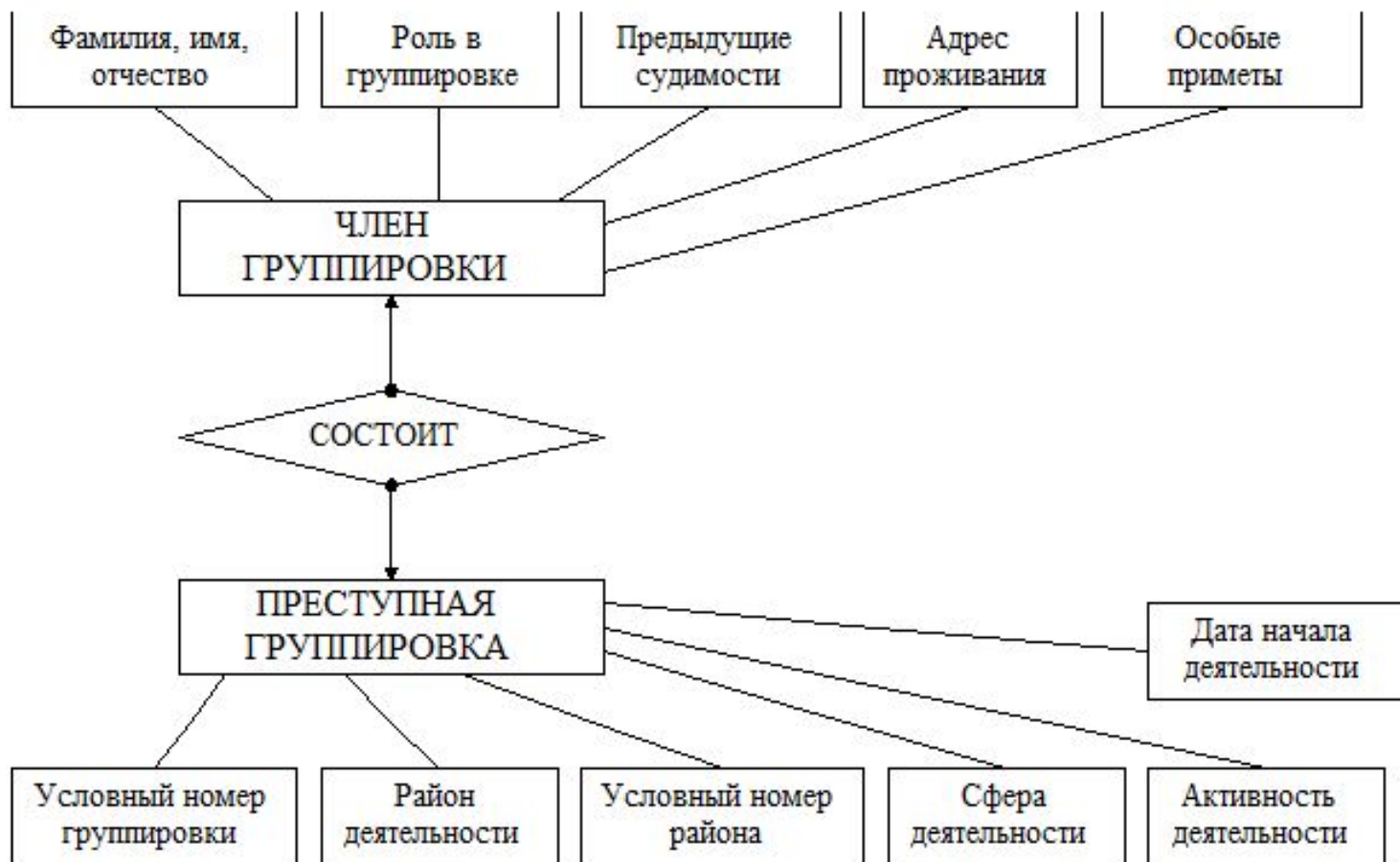


- **Пример.** Пусть в некотором локальном представлении выполняется описание преступных группировок. Предполагается, что член группировки может состоять только в одной группировке. При этом он может не иметь судимостей или иметь несколько. Преступная группировка может находиться в одном районе и иметь несколько сфер деятельности.



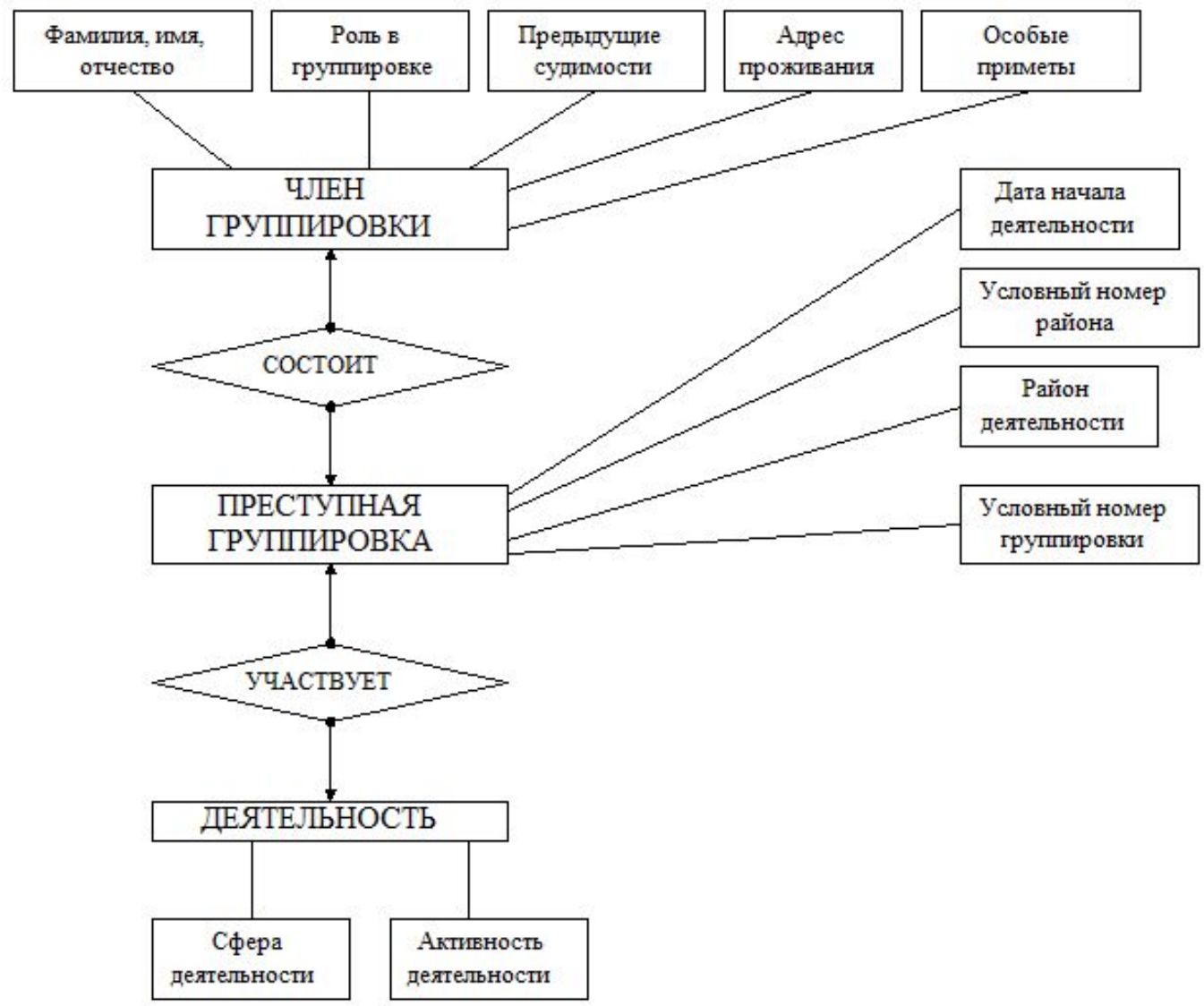
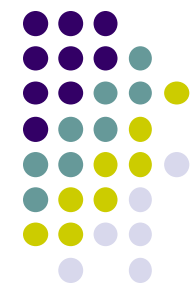


- С помощью данной графической диаграммы локального представления нельзя представить информации о членах группировки, не имеющих судимостей.
- Для решения этой задачи необходимо ввести в модель сущность ЧЛЕН ГРУППИРОВКИ, назначить ей соответствующие атрибуты, связать с сущностью ПРЕСТУПНАЯ ГРУППИРОВКА, если это необходимо, и удалить избыточные элементы.





- Полученный вариант не представляет информацию о группировках, если их активность в какой либо сфере деятельности не определена по разным причинам.
- Чтобы такую информацию можно было представлять, необходимо ввести в модель сущность **ДЕЯТЕЛЬНОСТЬ** и выполнить аналогичные процедуры построения, как и для сущности **ЧЛЕН ГРУППИРОВКИ**.





- База данных, реализующая это представление, окажется более гибкой в обработке данных, будет обладать большими возможностями по обработке произвольных запросов.
- Следовательно, для данного локального представления целесообразно ввести такие сущности, как ЧЛЕН ГРУППИРОВКИ, ПРЕСТУПНАЯ ГРУППИРОВКА, ДЕЯТЕЛЬНОСТЬ.



