

## Модели данных (1)

Ядром БД является *модель данных*.

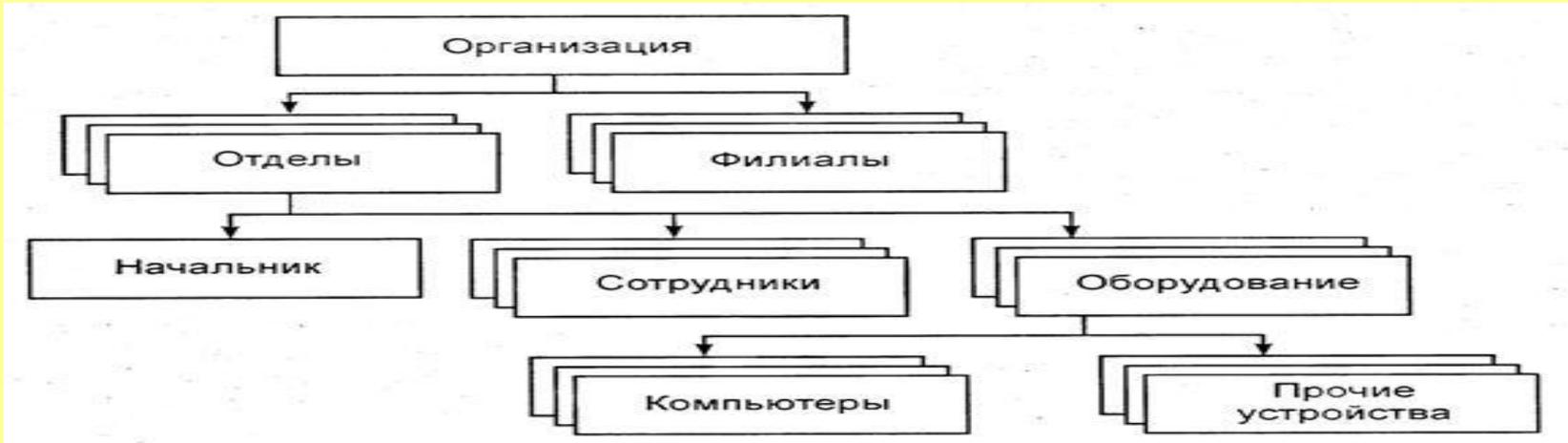
*Модель данных* – совокупность структур данных и операций их обработки.

Различают :

- иерархическую,
- сетевую,
- реляционную модели.

# Иерархическая модель

Иерархическая модель позволяет строить БД с древовидной структурой. В них каждый узел содержит свой тип данных (сущность). На верхнем уровне дерева имеется один узел – корень, на следующем уровне располагаются узлы, связанные с этим корнем, затем узлы, связанные с узлами предыдущего уровня, и т. д. Каждый узел может иметь только одного предка.

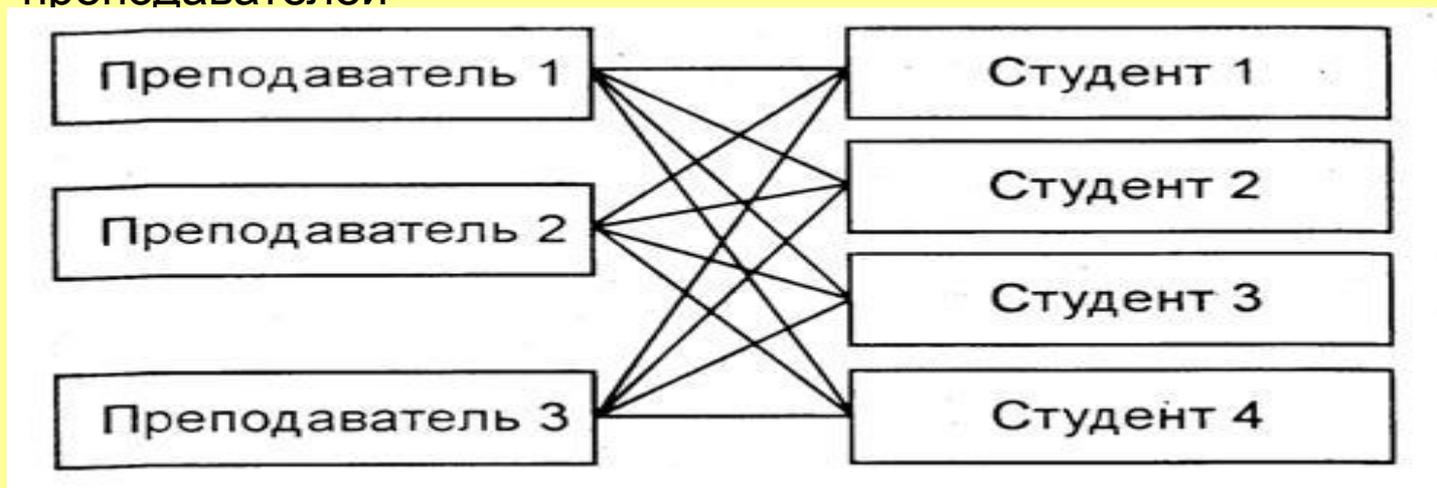


Поиск данных всегда начинается с корня. Затем производится спуск с одного уровня на другой пока не будет достигнут искомый уровень. Перемещения от одной записи к другой осуществляются с помощью ссылок.

Достоинствами иерархической модели являются простота описания иерархических структур реального мира, а также быстрое выполнение запросов, соответствующих структуре данных. Недостатки иерархической модели в том, что они часто содержат избыточные данные и не всегда удобно каждый раз начинать поиск нужных данных с корня.

## Сетевая модель

В *сетевой модели* возможны связи всех информационных объектов со всеми. Например, каждый преподаватель может обучать много студентов и каждый студент может обучаться у многих преподавателей



Использование иерархической и сетевой моделей ускоряет доступ к информации, но требует значительных ресурсов памяти, так как каждый элемент данных содержит ссылки на другие элементы. Характерна сложность реализации СУБД.

## Реляционная модель(РМД)

*Реляционная модель(РМД)* была разработана в начале 1970-х годов Эдгаром Ф. Коддом. В ней информация представляется в виде двумерных таблиц, а операции сводятся к манипуляциям с таблицами. В 1980-х годах она получила широкое распространение, а **реляционные СУБД стали промышленным стандартом.**

Причины доминирования РМД обусловлены тем, что имеются:

- развитая теория (реляционная алгебра);
- аппарат сведения других моделей данных к РМД;
- специальные средства ускоренного доступа к информации;
- стандартизированный высокоуровневый язык запросов к БД, позволяющий манипулировать данными без знания физической организации БД.

## Объектно-ориентированная модель

*Объектно-ориентированная модель* начала разрабатываться в 1990-е годы с появлением объектно-ориентированных языков. Такие БД хранят методы классов, что позволяет интегрировать данные и их обработку в приложениях.

## Основные понятия реляционной модели данных (1)

В математических дисциплинах понятию «таблица» соответствует понятие «отношение» (relation).

Таблица отражает объект реального мира – сущность, а каждая ее строка отражает конкретный экземпляр сущности.

Каждый столбец имеет уникальное для таблицы имя. Строки не имеют имен, порядок их следования не определен, а количество логически не ограничено. Одним из основных преимуществ РМД является однородность (каждая строка таблицы имеет один формат).

Пользователь сам решает вопрос, обладают ли соответствующие сущности однородностью. Этим решается проблема пригодности модели.



## Основные понятия реляционной модели данных (2)



Отношение представляет собой двумерную таблицу, содержащую некоторые данные. Сущность – объект любой природы, данные о котором хранятся в БД. Атрибуты – свойства, характеризующие сущность (столбцы). Степень отношения – количество столбцов. Схема отношения – список имен атрибутов, например, *СОТРУДНИК (№, ФИО, Год рождения, Должность, Кафедра)*. Домен – совокупность значений атрибутов отношения (тип данных). Кортеж – строка таблицы. Кардинальность (мощность) – количество строк в таблице.

## Основные понятия реляционной модели данных (3)



*Первичный ключ* – это атрибут, уникально идентифицирующий строки отношения. Первичный ключ из нескольких атрибутов называется составным. Первичный ключ не может быть полностью или частично пустым (иметь значение null). Ключи, которые можно использовать в качестве первичных, называются *потенциальными* или *альтернативными ключами*.

## Основные понятия реляционной модели данных (4)

*Внешний ключ* – это атрибут (атрибуты) одной таблицы, который может служить первичным ключом другой таблицы. Является ссылкой на первичный ключ другой таблицы



Отношения *СТУДЕНТ* (*ФИО, Группа, Специальность*) и *ПРЕДМЕТ* (*Назв Пр, Часы*) связаны отношением *СТУДЕНТ\_ПРЕДМЕТ* (*ФИО, Назв Пр, Оценка*), в котором внешние ключи *ФИО* и *Назв\_Пр* образуют составной ключ.

## Целостность реляционных данных (1)

Логические ограничения, накладываемые на данные, называются *ограничениями целостности*.

СУБД должна контролировать соответствие данных заданным ограничениям при переводе БД из одного состояния в другое. Использование ограничений связано также с адекватностью отражения предметной области.

*Явные ограничения* задаются семантикой предметной области. Они описывают области допустимых значений атрибутов, соотношение между атрибутами, динамику их изменения и т. д.

*Внутренние ограничения* свойственны собственно модели данных. Они накладываются на структуру отношений, на связи, на допустимые значения наборов данных, заложенные в выбранной модели данных. Способы реализации внутренних ограничений целостности зависят от СУБД.

## Целостность реляционных данных (2)

В РМД существует два вида внутренних ограничений целостности.

1. Целостность по существованию – потенциальный ключ отношения не может иметь пустого значения (NULL).
2. Целостность по связи – определяется понятием внешнего ключа отношения и означает систему правил, используемых для поддержания связей между записями в связанных таблицах. Обеспечивает защиту от случайного удаления или изменения связанных данных, от некорректного изменения ключевых полей

## Операции над отношениями (1)

РМД стала первой работоспособной моделью данных, поскольку имела эффективный инструментарий – операции реляционной алгебры. Основной единицей обработки является отношение, а не его кортежи. К отношениям можно применить *систему операций*, позволяющих получить одни отношения из других. Исключения составляют операции создания и заполнения таблиц, а также операции описания и переименования столбцов. Результатом запроса к реляционной БД может быть новое отношение, вычисленное на основе имеющихся отношений.

Реляционная алгебра включает две группы операций.

1. Традиционные операции над множествами (модифицированные с учетом того, что их операндами являются отношения) – объединение, пересечение, разность (вычитание), декартово произведение и деление.
2. Специальные реляционные операции – выборка, проекция, соединение.

## Операции над отношениями (2)

*Объединение* выполняется над двумя совместными отношениями  $R1, R2$  с идентичной структурой. В результате операции строится новое отношение  $R = R1 \cup R2$ , которое имеет тот же состав атрибутов и совокупность кортежей исходных отношений. В результирующее отношение по определению не включаются дубликаты кортежей. Ниже приведены исходные отношения:  $R1$  (табл. 2) и  $R2$  (табл. 3) и результат объединения –  $R$  (табл. 4).

Таблица 2

| ФИО          | Год рождения | Должность     | Кафедра |
|--------------|--------------|---------------|---------|
| Иванов И.И.  | 1948         | Зав. кафедрой | 22      |
| Сидоров С.С. | 1953         | Доцент        | 22      |
| Козлов К.К.  | 1980         | Ассистент     | 23      |

Таблица 3

| ФИО           | Год рождения | Должность         | Кафедра |
|---------------|--------------|-------------------|---------|
| Цветкова Н.Н. | 1965         | Доцент            | 23      |
| Петрова П.П.  | 1953         | Ст. преподаватель | 22      |
| Козлов К.К.   | 1980         | Ассистент         | 23      |

Таблица 4

| ФИО           | Год рождения | Должность         | Кафедра |
|---------------|--------------|-------------------|---------|
| Иванов И.И.   | 1948         | Зав. кафедрой     | 22      |
| Сидоров С.С.  | 1953         | Доцент            | 22      |
| Козлов К.К.   | 1980         | Ассистент         | 23      |
| Цветкова Н.Н. | 1965         | Доцент            | 23      |
| Петрова П.П.  | 1953         | Ст. преподаватель | 22      |

## Операции над отношениями (3)

*Пересечение* выполняется над двумя совместными отношениями  $R1, R2$ . Результирующее отношение  $RP = R1 \cap R2$  содержит кортежи, которые есть в каждом из исходных. Результат имеет тот же состав атрибутов, что и исходные отношения. Пересечение отношений  $R1$  и  $R2$  дает отношение  $RP$  (табл. 5).

Таблица 5

| ФИО         | Год рождения | Должность | Кафедра |
|-------------|--------------|-----------|---------|
| Козлов К.К. | 1980         | Ассистент | 23      |

*Вычитание* выполняется над двумя совместными отношениями  $R1, R2$ . В результате строится новое отношение  $RV = R1 - R2$  с идентичным набором атрибутов, содержащее кортежи первого отношения  $R1$ , которые не входят в отношение  $R2$ . Вычитание отношения  $R2$  из  $R1$  дает отношение  $RV$  (табл. 6).

Таблица 6

| ФИО          | Год рождения | Должность     | Кафедра |
|--------------|--------------|---------------|---------|
| Иванов И.И.  | 1948         | Зав. кафедрой | 22      |
| Сидоров С.С. | 1953         | Доцент        | 22      |

## Операции над отношениями (4)

*Декартово произведение* выполняется над двумя отношениями  $R1$  и  $R2$ , имеющими в общем случае разный состав атрибутов. В результате образуется новое отношение  $RD = R1 \times R2$ , которое включает все атрибуты исходных отношений. Результирующее отношение состоит из всевозможных сочетаний кортежей исходных отношений. Число кортежей (мощность) отношения-произведения равно произведению мощностей исходных отношений.

Декартово произведение отношений  $R1$  (табл. 7) и  $R2$  (табл. 8) дает новое отношение  $RD$  (табл. 9), которое содержит все атрибуты исходных отношений. В него целесообразно добавить атрибут *Оценка* для записи результатов экзамена.

Таблица 7

| Номер студента | ФИО студента |
|----------------|--------------|
| 11             | Иванов И.И.  |
| 12             | Петров П.П.  |
| 13             | Сидоров С.С. |

Таблица 8

| Код дисциплины | Наименование |
|----------------|--------------|
| Д1             | Математика   |
| Д2             | Информатика  |

Таблица 9

| Номер студента | ФИО студента | Код дисциплины | Наименование | Оценка |
|----------------|--------------|----------------|--------------|--------|
| 11             | Иванов И.И.  | Д1             | Математика   | 5      |
| 12             | Петров П.П.  | Д1             | Математика   | 3      |
| 13             | Сидоров С.С. | Д1             | Математика   | 5      |
| 11             | Иванов И.И.  | Д2             | Информатика  | 5      |
| 12             | Петров П.П.  | Д2             | Информатика  | 4      |
| 13             | Сидоров С.С. | Д2             | Информатика  | 4      |

## Операции над отношениями (5)

*Деление* выполняется над двумя отношениями  $R1$  и  $R2$ , имеющими в общем случае разные структуры и часть одинаковых атрибутов. В результате образуется новое отношение, содержащее атрибуты 1-го операнда, отсутствующие во 2-м операнде, и кортежи 1-го операнда, которые совпали с кортежами 2-го. Для выполнения этой операции 2-й операнд должен содержать лишь атрибуты, совпадающие с атрибутами 1-го.

Например, чтобы узнать, кто из студентов получил по математике 5 и по информатике 4, надо разделить отношения *Экзаменационная ведомость* на вспомогательное отношение *Мат5Физ4* (Наименование, Оценка) с двумя кортежами: *Математика, 5* и *Информатика, 4*. В результате получим отношение *Итог* (Номер студента, ФИО студента, Код дисциплины) с одним кортежем – *13, Сидоров, Д1*.

*Выборка* выполняется над одним отношением  $R$ . Для отношения по заданному условию (предикату) осуществляется выборка подмножества кортежей. Результирующее отношение имеет ту же структуру, что и исходное, но число его кортежей будет меньше (или равно) числа кортежей исходного отношения. Например, выбрать студентов, сдавших математику на отлично (*Код дисциплины = Д1*) AND (*Оценка = 5*) (табл. 10).

Таблица 10

| Номер студента | ФИО студента | Код дисциплины | Наименование | Оценка |
|----------------|--------------|----------------|--------------|--------|
| 11             | Иванов И.И.  | Д1             | Математика   | 5      |
| 13             | Сидоров С.С. | Д1             | Математика   | 5      |

## Операции над отношениями (6)

*Проекция* выполняется над одним отношением  $R$ . Операция формирует новое отношение  $RPR$  с заданным подмножеством атрибутов исходного отношения  $R$ . Оно может содержать меньше кортежей, так как после отбрасывания в исходном отношении  $R$  части атрибутов (и возможного исключения первичного ключа) могут образоваться кортежи-дубли, которые из результирующего отношения исключаются по определению.

Ниже приведен пример исходного отношения  $R$  (табл. 11) и результат проекции этого отношения на два его атрибута - *Должность* и *Номер отдела* (табл. 12).

Таблица 11

| ФИО           | Номер<br>отдела | Должность |  | Номер<br>отдела | Должность |
|---------------|-----------------|-----------|--|-----------------|-----------|
| Иванов И.И.   | 01              | Инженер   |  | 01              | Инженер   |
| Петров П.П.   | 02              | Инженер   |  | 02              | Инженер   |
| Нестеров Н.Н. | 01              | Инженер   |  | 03              | Лаборант  |
| Никитин К.К.  | 02              | Лаборант  |  |                 |           |

Таблица 12

*Соединение* выполняется для заданного условия соединения над двумя логически связанными отношениями. Исходные отношения  $R1$  и  $R2$  имеют разные структуры, в которых есть одинаковые атрибуты – внешние ключи. Операция соединения формирует новое отношение, структура которого является совокупностью всех атрибутов исходных отношений. Результирующие кортежи формируются соединением каждого кортежа из  $R1$  с теми кортежами  $R2$ , для которых выполняется условие соединения. В зависимости от этого условия соединение называется: *естественным* – равенство значений общих атрибутов отношений  $R1$  и  $R2$ ; *эквисоединением* – равенство значений атрибутов, входящих в условие соединения; *тета-соединением* – другой знак сравнения.

Операция соединения имеет большое значение для РБД, так как в процессе нормализации отношений исходное отношение разбивается на несколько более мелких отношений, которые при выполнении запросов пользователя требуется, как правило, вновь соединять для восстановления исходного отношения.

Рассмотренные выше операции в той или иной мере реализуются в языке манипулирования данными СУБД (SQL, QBE, другие языки запросов). Язык SQL является более чем реляционно-полным, так как кроме операций реляционной алгебры содержит полный набор операторов над кортежами – *Включить, Удалить, Изменить*, а также реализует арифметические операции и операции сравнения.

К достоинствам РМД относятся:

- простота представления данных благодаря табличной форме;
- минимальная избыточность данных при нормализации отношений;
- обеспечение независимости приложений пользователя от данных, допускающей включение или удаление отношений, изменение их атрибутивного состава;

К недостаткам РМД можно отнести то, что нормализация данных приводит к значительной их фрагментации, в то время как в большинстве задач необходимо объединение фрагментированных данных.

# Нормализация баз данных (1)

Данные могут группироваться в таблицы (отношения) разными способами. При проектировании БД в качестве отправной точки может использоваться одно *универсальное отношение*, в которое включаются все необходимые атрибуты. Оно может содержать все данные, которые предполагается размещать в БД.

В качестве примера рассмотрим универсальное отношение *сотрудники*, содержащее информацию о сотрудниках предприятия (табл. 13).

Таблица 13

| Код сотрудника | ФИО            | Должность               | Номер отдела | Наименование отдела  | Квалификация   |
|----------------|----------------|-------------------------|--------------|----------------------|----------------|
| 7513           | Иванов И.И.    | Программист             | 128          | Отдел проектирования | C, Java        |
| 9842           | Сергеева С.С.  | Администратор БД        | 42           | Финансовый отдел     | DB2            |
| 6651           | Петров П.П.    | Программист             | 128          | Отдел проектирования | VB, Java       |
| 9006           | Николаев Н. Н. | Системный администратор | 128          | Отдел проектирования | Windows, Linux |

При использовании универсального отношения возникают две проблемы:

- избыточность данных;
- потенциальная противоречивость (аномалии).

**Под избыточностью** понимают повторение данных в разных строках одной таблицы или в разных таблицах БД. Так, для каждого сотрудника отдела 128 повторяются данные «128, Отдел проектирования».

**Аномалии** – это проблемы, возникающие в данных из-за дефектов проектирования БД.

Существуют три вида аномалий: вставки, удаления и модификации.

**Аномалии вставки** проявляются при вводе данных в дефектную таблицу. Добавляя информацию о новом сотруднике, мы должны добавить номер и название отдела. Если ввести данные, не соответствующие имеющимся в таблице (например, 42, отдел проектирования), будет не ясно, какая из строк БД содержит правильную информацию

**Аномалии удаления** возникают при удалении данных из дефектной схемы. Предположим, что все сотрудники отдела 128 уволились в один и тот же день. После удаления записей этих сотрудников в БД больше не будет ни одной записи, содержащей информацию об отделе 128.

**Аномалии модификации** возникают при изменении данных дефектной схемы. Предположим, что отдел 128 решили переименовать в отдел передовых технологий. Необходимо изменить соответствующие данные о каждом сотруднике отдела. Если мы

## Нормализация баз данных (2)

Правилом разработки хорошей структуры БД является необходимость избегать схем с большим числом *пустых атрибутов*. Если мы хотим указать, что один из ста служащих имеет особую квалификацию, для хранения этой информации не следует добавлять в таблицу еще один столбец, поскольку для остальных 99 работников значением столбца будет NULL. Вместо этого следует добавить новую таблицу, в которой будут храниться только кодовые номера и информация о квалификации тех работников, которых это касается.

Решение перечисленных проблем состоит в разделении данных и связей, что обеспечивается процедурой **нормализации**. Концепции и методы нормализации были разработаны Э. Ф. Коддом.

**Нормализация отношений** – это формальный аппарат ограничений на формирование отношений, который позволяет устранить дублирование и потенциальную противоречивость хранимых данных, уменьшает трудозатраты на ведение БД. Процесс нормализации заключается в декомпозиции исходных отношений на более простые отношения. Цель нормализации – получение такого проекта БД, в котором «каждый факт появляется лишь в одном месте».

## Нормализация баз данных (3)

Теория нормализации основана на наличии зависимостей между атрибутами отношения. Основными видами зависимостей являются:

- **функциональные;**
- **многозначные;**
- **транзитивные.**

Базовым является понятие **функциональной зависимости**, поскольку на его основе формируются определения всех остальных видов зависимостей. Атрибут В функционально зависит от атрибута А, если каждому значению А соответствует в точности одно значение В.

Математически функциональную зависимость В от А обозначают  $A \rightarrow B$ . Это означает, что во всех кортежах с одинаковым значением атрибута А атрибут В будет иметь также одно и то же значение. При этом А и В могут быть составными, то есть состоять из двух и более атрибутов.

Зависимость, при которой каждый неключевой атрибут зависит от всего составного ключа и не зависит от его частей, называется **полной функциональной зависимостью**. Если атрибут А зависит от атрибута В, а атрибут В зависит от атрибута С ( $C \rightarrow B \rightarrow A$ ), но обратная зависимость отсутствует, то зависимость А от С называется **транзитивной**.

## Нормализация баз данных (4)

**Многозначная зависимость.** Говорят, что один атрибут отношения многозначно определяет другой атрибут того же отношения, если для каждого значения первого атрибута существует множество соответствующих значений второго атрибута. Многозначные зависимости могут быть:

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

## Нормализация баз данных (5)

Каждая ступень процесса нормализации приводит схему отношений в последовательные нормальные формы. Для каждой ступени имеются наборы ограничений. Выделяют следующую последовательность нормальных форм:

- первая нормальная форма (1НФ);
- вторая нормальная форма (2НФ);
- третья нормальная форма (3НФ);
- усиленная 3НФ или нормальная форма Бойса-Кодда (БКНФ);
- четвертая нормальная форма (4НФ);
- пятая нормальная форма (5НФ).

## Нормализация баз данных (6)

Отношение находится в *первой нормальной форме (1НФ)*, когда каждая строка содержит только одно значение для каждого атрибута (столбца), то есть все атрибуты отношения имеют единственное значение (являются атомарными).

В столбце *Квалификация* ненормализованной табл. 13 содержатся списки значений (C, Java и т. д.). Чтобы привести схему к 1НФ, необходимо разместить в этом столбце атомарные значения. Самый простой способ заключается в выделении по одной строке на каждый элемент квалификации (табл. 14).

Таблица 14

| Код сотрудника | ФИО            | Должность               | Номер отдела | Наименование отдела  | Квалификация |
|----------------|----------------|-------------------------|--------------|----------------------|--------------|
| 7513           | Иванов И.И.    | Программист             | 128          | Отдел проектирования | C            |
| 7513           | Иванов И.И.    | Программист             | 128          | Отдел проектирования | Java         |
| 9842           | Сергеева С.С.  | Администратор БД        | 42           | Финансовый отдел     | DB2          |
| 6651           | Петров П.П.    | Программист             | 128          | Отдел проектирования | VB           |
| 6651           | Петров П.П.    | Программист             | 128          | Отдел проектирования | Java         |
| 9006           | Николаев Н. Н. | Системный администратор | 128          | Отдел проектирования | Windows      |
| 9006           | Николаев Н. Н. | Системный администратор | 128          | Отдел проектирования | Linux        |

Такое решение далеко от идеального, поскольку порождает очевидную избыточность данных – для каждой комбинации сотрудник-квалификация приходится хранить все характеристики сотрудника.

## Нормализация баз данных (7)

Отношение находится во второй нормальной форме (2НФ), если оно находится в 1НФ, и каждый неключевой атрибут полностью функционально зависит от всех составляющих первичного ключа. Если атрибут не зависит полностью от первичного ключа, то он внесен ошибочно и должен быть удален. Нормализация производится путем нахождения существующего отношения, к которому относится данный атрибут, или созданием нового отношения, в который атрибут должен быть помещен.

Таблица *Квалификации\_сотрудников* (табл. 14) находится в 1НФ, но не удовлетворяет 2НФ. Первичный ключ должен уникальным образом идентифицировать каждую строку. Единственным вариантом является использование в качестве первичного ключа комбинации *Код сотрудника* и *Квалификация*. Это порождает схему: *Квалификации\_сотрудников* (*Код сотрудника, ФИО, Должность, Номер отдела, Наименование отдела, Квалификация*).

Одной из имеющихся здесь функциональных зависимостей будет: *Код сотрудника, Квалификация* ' *ФИО, Должность, Номер отдела, Наименование отдела*. Но, кроме того, мы также имеем: *Код сотрудника* ' *ФИО, Должность, Номер отдела, Наименование отдела*. Другими словами, можно определить имя, должность и отдел, используя только код сотрудника. Это значит, что указанные атрибуты функционально зависимы только от части первичного ключа, а не от всего первичного ключа. Следовательно, указанная схема не находится в 2НФ.

## Нормализация баз данных (8)

Для приведения этой схемы в 2НФ необходимо декомпозировать исходное отношение на два, в которых все неключевые атрибуты будут полностью функционально зависеть от ключа: *сотрудники* (Код сотрудника, ФИО, Должность, Номер отдела, Наименование отдела) и *Квалификации\_сотрудников* (Код сотрудника, Квалификация) (табл. 15–16).

Таблица 15

| Код сотрудника | ФИО           | Должность               | Номер отдела | Наименование отдела  |
|----------------|---------------|-------------------------|--------------|----------------------|
| 7513           | Иванов И.И.   | Программист             | 128          | Отдел проектирования |
| 9842           | Сергеева С.С. | Администратор БД        | 42           | Финансовый отдел     |
| 6651           | Петров П.П.   | Программист             | 128          | Отдел проектирования |
| 9006           | Николаев Н.Н. | Системный администратор | 128          | Отдел проектирования |

Таблица 16

| Код сотрудника | Квалификация |
|----------------|--------------|
| 7513           | C            |
| 7513           | Java         |
| 9842           | DB2          |
| 6651           | VB           |
| 6651           | Java         |
| 9006           | Windows      |
| 9006           | Linux        |

## Нормализация баз данных (8)

Отношение находится в *третьей нормальной форме (3НФ)*, если оно находится во 2НФ и ни один из его неключевых атрибутов не связан функциональной зависимостью с любым другим неключевым атрибутом. Атрибуты, зависящие от других неключевых атрибутов, нормализуются путем перемещения зависимого атрибута и атрибута, от которого он зависит, в новое отношение.

Формально, для приведения схемы в 3НФ необходимо исключить все транзитивные зависимости. Схема отношения *сотрудники* (табл. 15) содержит следующие функциональные зависимости: *Код сотрудника ' ФИО, Должность, Номер отдела, Наименование отдела и Номер отдела ' Наименование отдела*.

Первичным ключом является *Код сотрудника*, и все атрибуты полностью функционально зависимы от него (первичный ключ определяется единственным атрибутом). При этом *Номер отдела* ключом не является.

## Нормализация баз данных (9)

Функциональная зависимость *Код сотрудника ' Наименование отдела* является транзитивной, поскольку содержит промежуточный шаг (зависимость *Номер отдела ' Наименование отдела*). Для приведения в 3НФ необходимо исключить эту транзитивную зависимость, декомпозируя отношение на два: *сотрудники* (*Код сотрудника, ФИО, Должность, Номер отдела*) и *отделы* (*Номер отдела, Наименование отдела*) (табл. 17–18).

Таблица 17

| Код сотрудника | ФИО           | Должность               | Номер отдела |
|----------------|---------------|-------------------------|--------------|
| 7513           | Иванов И.И.   | Программист             | 128          |
| 9842           | Сергеева С.С. | Администратор БД        | 42           |
| 6651           | Петров П.П.   | Программист             | 128          |
| 9006           | Николаев Н.Н. | Системный администратор | 128          |

Таблица 18

| Номер отдела | Наименование отдела  |
|--------------|----------------------|
| 42           | Финансовый отдел     |
| 128          | Отдел проектирования |

# Нормализация баз данных (10)

Нормальная форма Бойса-Кодда (БКНФ) является развитием 3НФ и требует, чтобы в отношении были только такие функциональные зависимости, левая часть которых является потенциальным ключом отношения. *Потенциальный ключ* представляет собой атрибут (или множество атрибутов), который может быть использован для данного отношения в качестве первичного ключа. Фактически первичный ключ – это один из потенциальных ключей, назначенный в качестве первичного. *Детерминантом* называется левая часть функциональной зависимости. Отношение находится в БКНФ тогда и только тогда, когда каждый детерминант отношения является потенциальным ключом.

Алгоритм приведения ненормализованных схем в 3НФ показан на рис. 15. На практике построение 3НФ в большинстве случаев является достаточным и приведением к ней процесс построения реляционной БД заканчивается.

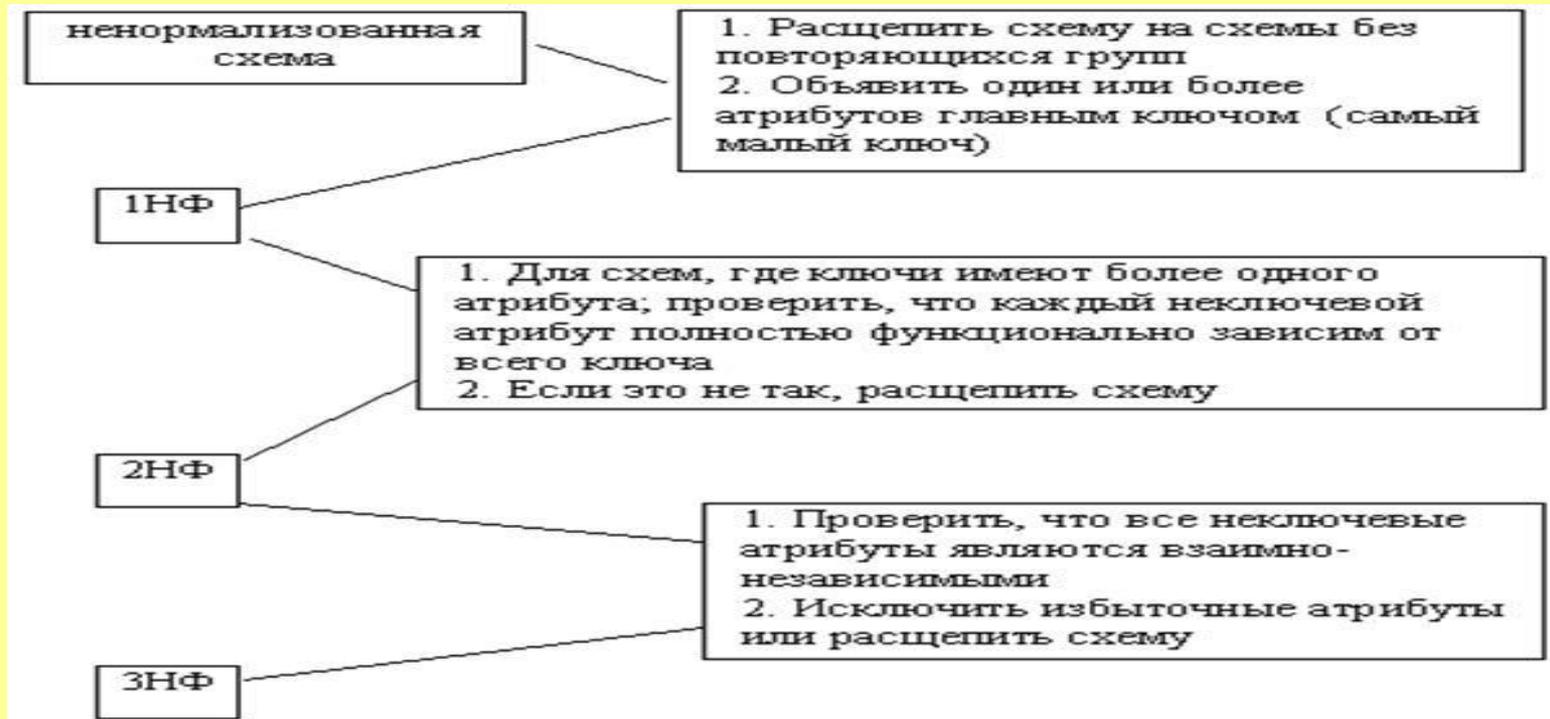


Рис. 15. Алгоритм приведения ненормализованных схем в 3НФ

**Запомнить правила нормализации помогает изречение:  
«Нормализация – это ключ, целый ключ и ничего, кроме ключа».**

Нормальные формы высших порядков (4НФ и 5НФ) представляют большой интерес для теоретических исследований, чем для практики проектирования БД. В них учитываются многозначные зависимости между атрибутами. Полной декомпозицией отношения называют такую совокупность произвольного числа его проекций, соединение которых позволяет получить исходное отношение.

Отношение находится в пятой нормальной форме (5НФ), когда в каждой его полной декомпозиции все проекции содержат возможный ключ.

Отношение, не имеющее ни одной полной декомпозиции, также находится в 5НФ.

Четвертая нормальная форма (4НФ) является частным случаем 5НФ, когда полная декомпозиция должна быть соединением ровно двух проекций. На практике непросто подобрать отношение, которое находится в 4НФ, не будучи в 5НФ.

Нормализация – это процесс последовательной замены отношения его полными декомпозициями до тех пор, пока все они не будут находиться в 5НФ. Приведя отношения к БКНФ, можно с большой гарантией считать, что они находятся в 5НФ. Единственными функциональными зависимостями в любом отношении должны быть зависимости вида  $K' \rightarrow A$ , где  $K$  – первичный ключ, а  $A$  – атрибут.