

---

# Проектирование РБД на основе учета функциональных зависимостей

---

С.Д. Кузнецов. Базы данных. Тема 5

# План (1)

- Введение
- Элементы теории функциональных зависимостей
  - Базовые определения и утверждения теории функциональных зависимостей
    - ✓ Общие определения
    - ✓ Замыкание множества функциональных зависимостей. Аксиомы Армстронга. Замыкание множества атрибутов
    - ✓ Минимальное покрытие множества функциональных зависимостей
  - Декомпозиция без потерь и функциональные зависимости
    - ✓ Корректные и некорректные декомпозиции отношений. Теорема Хита
    - ✓ Диаграммы функциональных зависимостей

# План (2)

- Минимальные функциональные зависимости и вторая нормальная форма
  - Аномалии обновления, возникающие из-за наличия не минимальных функциональных зависимостей
  - Возможная декомпозиция
  - Вторая нормальная форма
- Нетранзитивные функциональные зависимости и третья нормальная форма
  - Аномалии обновления, возникающие из-за наличия транзитивных функциональных зависимостей
  - Возможная декомпозиция
  - Третья нормальная форма
  - Независимые проекции отношений. Теорема Риссанена

# План (3)

- Перекрывающиеся возможные ключи и нормальная форма Бойса-Кодда
  - Аномалии обновлений, связанные с наличием перекрывающихся возможных ключей
  - Нормальная форма Бойса-Кодда
  - Всегда ли следует стремиться к BCNF?

# Введение (1)

- При проектировании баз данных решаются две основные проблемы:
  - Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области и было, по возможности, наилучшим (эффективным, удобным и т. д.)? Часто эту проблему называют проблемой логического проектирования баз данных
  - Как обеспечить эффективность выполнения запросов к базе данных, т. е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создания каких дополнительных структур (например, индексов) потребовать и т. д.? Эту проблему обычно называют проблемой физического проектирования баз данных
- В случае реляционных баз данных трудно предложить какие-либо общие рецепты по части физического проектирования
- Здесь слишком многое зависит от используемой СУБД
- Поэтому мы ограничимся вопросами логического проектирования реляционных баз данных, которые существенны при использовании любой реляционной СУБД.

# Введение (2)

- Более того, мы не будем касаться очень важного аспекта проектирования – определения ограничений целостности общего вида (за исключением ограничений, задаваемых функциональными и многозначными зависимостями, а также зависимостями проекции/соединения)
- Дело в том, что при использовании СУБД с развитыми механизмами определения и поддержки ограничений целостности (например, SQL-ориентированных систем) трудно предложить какой-либо универсальный подход к определению ограничений целостности
- Эти ограничения могут иметь произвольно сложную форму, и их формулировка пока относится скорее к области искусства, чем инженерного мастерства
- Самое большее, что предлагается по этому поводу в литературе, это автоматическая проверка непротиворечивости набора ограничений целостности.

# Введение (3)

- Поэтому мы будем считать, что проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том, из каких отношений должна состоять БД и какие атрибуты должны быть у этих отношений
- Будет рассмотрен классический подход, при котором весь процесс проектирования базы данных осуществляется в терминах реляционной модели данных методом последовательных приближений к удовлетворительному набору схем отношений
- Исходной точкой является представление предметной области в виде одного или нескольких отношений, и на каждом шаге проектирования производится некоторый набор схем отношений, обладающих «улучшенными» свойствами
- Процесс проектирования представляет собой процесс нормализации схем отношений, причем каждая следующая нормальная форма обладает свойствами, в некотором смысле, лучшими, чем предыдущая.

# Введение (4)

- Каждой нормальной форме соответствует определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений
- Примером может служить ограничение первой нормальной формы – значения всех атрибутов отношения атомарны
- Поскольку требование первой нормальной формы является базовым требованием классической реляционной модели данных, мы будем считать, что исходный набор отношений уже соответствует этому требованию
- В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:
  - 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)

# Введение (5)

- Основные свойства нормальных форм состоят в следующем:
  - каждая следующая нормальная форма в некотором смысле лучше предыдущей нормальной формы;
  - при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются
- В основе процесса проектирования лежит метод нормализации, т. е. декомпозиции отношения, находящегося в предыдущей нормальной форме, на два или более отношений, которые удовлетворяют требованиям следующей нормальной формы.
- В этой лекции мы обсудим первые шаги процесса нормализации, в которых учитываются функциональные зависимости между атрибутами отношений
- Хотя мы и называем эти шаги первыми, именно они имеют основную практическую важность, поскольку позволяют получить схему реляционной базы данных, в большинстве случаев удовлетворяющую потребности приложений

# Элементы теории функциональных зависимостей (1)

- Этот курс не посвящен подробному описанию основных результатов в области теории реляционных баз данных
- Обеспечиваются только определения и утверждения, необходимые для общего понимания процесса проектирования реляционных баз данных на основе нормализации.
- Поскольку наиболее важные с практической точки зрения свойства реляционных баз данных базируются на понятии *функциональной зависимости*, мы выделили в отдельный раздел краткое обсуждение соответствующих теоретических вопросов
- Среди этих вопросов наибольший интерес представляют
  - замыкания и покрытия множеств функциональных зависимостей,
  - аксиомы Армстронга и
  - теорема Хита о достаточном условии декомпозиции отношения без потерь.
- Понятия и утверждения данного раздела действительно нужны для усвоения материала этой темы, но мы стремились еще и продемонстрировать читателям на несложных примерах, что собой представляет теория реляционных баз данных, каков уровень ее сложности и насколько она понятна интуитивно

# Элементы теории функциональных зависимостей (2)

## Базовые определения и утверждения (1)

- Пусть задана переменная отношения  $r$ , и  $X$  и  $Y$  являются произвольными подмножествами заголовка  $r$  («составными» атрибутами)
- **Определение 5.1. Функциональная зависимость**  
В значении переменной отношения  $r$  атрибут  $Y$  *функционально зависит* от атрибута  $X$  в том и только в том случае, если каждому значению  $X$  соответствует в точности одно значение  $Y$ 
  - В этом случае говорят также, что атрибут  $X$  *функционально определяет* атрибут  $Y$  ( $X$  является *детерминантом* (*определителем*) для  $Y$ , а  $Y$  является зависимым от  $X$ )
  - Будем обозначать это как  $r.X \rightarrow r.Y$

# Элементы теории функциональных зависимостей (3)

## Базовые определения и утверждения (2)

- Для примера будем использовать переменную отношения СЛУЖАЩИЕ\_ПРОЕКТЫ с заголовком {СЛУ\_НОМ, СЛУ\_ИМЯ, СЛУ\_ЗАРП, ПРО\_НОМ, ПРОЕКТ\_РУК}
- Очевидно, что если первичным ключом переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ является СЛУ\_НОМ, то для этого отношения справедлива, например, функциональная зависимость (Functional Dependency – FD) СЛУ\_НОМ →

СЛУ\_ИМЯ

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
4434	Иванов	22400.00	1	Иванов
4435	Петров	29000.00	1	Иванов
4415	Сидоров	23000.00	1	Иванов
4436	Федоров	20600.00	1	Иванов
4440	Иванова	22000.00	1	Иванов
4441	Сидоренко	18000.00	2	Иваненко
4416	Федоренко	20400.00	2	Иваненко
4417	Иваненко	21600.00	2	Иваненко

# Элементы теории функциональных зависимостей (4)

## Базовые определения и утверждения (3)

- На самом деле, для этого значения переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ выполняются еще и следующие FD (1):
  - СЛУ\_НОМ → СЛУ\_ЗАРП
  - СЛУ\_НОМ → ПРО\_НОМ
  - СЛУ\_НОМ → ПРОЕКТ\_РУК
  - {СЛУ\_НОМ, СЛУ\_ИМЯ} → СЛУ\_ЗАРП
  - {СЛУ\_НОМ, СЛУ\_ИМЯ} → ПРО\_НОМ
  - {СЛУ\_НОМ, СЛУ\_ИМЯ} → {СЛУ\_ЗАРП, ПРО\_НОМ}
  - ...
  - ПРО\_НОМ → ПРОЕКТ\_РУК и т.д.
- Поскольку имена всех служащих различны, то выполняются и такие FD (2):
  - СЛУ\_ИМЯ → СЛУ\_НОМ
  - СЛУ\_ИМЯ → СЛУ\_ЗАРП
  - СЛУ\_ИМЯ → ПРО\_НОМ и т.д.
- Более того, для этого значения отношения выполняется и FD (3):
  - СЛУ\_ЗАРП → ПРО\_НОМ

# Элементы теории функциональных зависимостей (5)

## Базовые определения и утверждения (4)

- Однако заметим, что природа FD группы (1) отличается от природы FD групп (2) и (3).
  - Логично предположить, что идентификационные номера служащих должны быть всегда различны, а у каждого проекта имеется только один руководитель.
  - Поэтому FD группы (1) должны быть верны для любого допустимого значения переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ и могут рассматриваться как *инварианты*, или *ограничения целостности* этой переменной отношения.
- FD группы (2) базируются на менее естественном предположении о том, что имена всех служащих различны
  - Это соответствует действительности для данного возможного значения отношения, но возможно, что с течением времени FD группы (2) не будут выполняться для какого-либо значения переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ.
- Наконец, FD группы (3) основана на совсем неестественном предположении, что никакие двое служащих, участвующие в разных проектах, не получают одинаковую зарплату
  - Опять же, данное предположение верно для возможного значения отношения, но, скорее всего, это случайное совпадение.

# Элементы теории функциональных зависимостей (6)

## Базовые определения и утверждения (5)

- В дальнейшем нас будут интересовать только те функциональные зависимости, которые должны выполняться для всех возможных значений переменных отношений
- Заметим, что если атрибут  $A$  переменной отношения  $r$  является *возможным ключом*, то для любого атрибута  $B$  этого отношения всегда выполняется FD  $A \rightarrow B$ 
  - в группе (1) к этим FD относятся все FD, детерминантом которых является атрибут СЛУ\_НОМ).
- Обратите внимание, что наличие в отношении СЛУЖАЩИЕ\_ПРОЕКТЫ FD ПРО\_НОМ  $\rightarrow$  ПРОЕКТ\_РУК приводит к некоторой *избыточности* этого отношения
- Имя руководителя проекта является характеристикой проекта, а не служащего, но в нашем случае содержится в теле отношения столько раз, сколько служащих работает над проектом.
- Итак, мы будем иметь дело с FD, которые выполняются для всех возможных состояний тела соответствующего отношения и могут рассматриваться как ограничения целостности
- Как показывает (неполный) список (1), таких зависимостей может быть очень много
- Поскольку они трактуются как ограничения целостности, за их соблюдением должна следить СУБД
- Поэтому важно уметь сократить набор FD до минимума, поддержка которого гарантирует выполнение всех зависимостей

# Элементы теории функциональных зависимостей (7)

## Базовые определения и утверждения (6)

### ■ **Определение 5.2. Тривиальная функциональная зависимость**

FD  $A \rightarrow B$  называется тривиальной, если  $A \supseteq B$

➤ т. е. множество атрибутов  $A$  включает множество  $B$  или совпадает с множеством  $B$

■ Очевидно, что любая тривиальная FD всегда выполняется

■ Например, в отношении СЛУЖАЩИЕ\_ПРОЕКТЫ всегда выполняется FD

$\{\text{СЛУ\_ЗАРП, ПРО\_НОМ}\} \rightarrow \text{СЛУ\_ЗАРП}$

■ Частным случаем тривиальной FD является  $A \rightarrow A$

■ Поскольку тривиальные FD выполняются всегда, их нельзя трактовать как ограничения целостности, и поэтому они не представляют интереса с практической точки зрения

■ Однако в теоретических рассуждениях их наличие необходимо учитывать

## Элементы теории функциональных зависимостей (8)

### Базовые определения и утверждения (7)

#### ■ **Определение 5.3. Замыкание множества FD**

Замыканием множества FD  $S$  является множество FD  $S^+$ , включающее все FD, логически выводимые из FD множества  $S$

■ Для начала приведем два примера FD, из которых следуют (или *выводятся*) другие FD

■ Будем снова пользоваться отношением СЛУЖАЩИЕ\_ПРОЕКТЫ

■ Для этого отношения выполняется, например, FD

$СЛУ\_НОМ \rightarrow \{СЛУ\_ЗАРП, ОТД\_НОМ\}$

➤ Из нее выводятся FD

$СЛУ\_НОМ \rightarrow СЛУ\_ЗАРП$  и  $СЛУ\_НОМ \rightarrow ОТД\_НОМ$

## Элементы теории функциональных зависимостей (9)

### Базовые определения и утверждения (8)

- В этом отношении СЛУЖАЩИЕ\_ПРОЕКТЫ имеется также пара FD  $СЛУ\_НОМ \rightarrow ОТД\_НОМ$  и  $ОТД\_НОМ \rightarrow ПРОЕКТ\_РУК$ 
  - Из них выводится FD  $СЛУ\_НОМ \rightarrow ПРОЕКТ\_РУК$
  - FD вида  $СЛУ\_НОМ \rightarrow ПРОЕКТ\_РУК$  называются *транзитивными*, поскольку ПРОЕКТ\_РУК зависит от СЛУ\_НОМ «транзитивно», через атрибут ПРО\_НОМ
- **Определение 5.4. Транзитивная функциональная зависимость**  
FD  $A \rightarrow C$  называется транзитивной, если существует такой атрибут  $B$ , что имеются функциональные зависимости  $A \rightarrow B$  и  $B \rightarrow C$  и отсутствует функциональная зависимость  $C \rightarrow A$ .

## Элементы теории функциональных зависимостей (8)

### Базовые определения и утверждения (7)

- Подход к решению проблемы поиска замыкания  $S^+$  множества FD  $S$  впервые предложил Вильям Армстронг
- Им был предложен набор *правил вывода* новых FD из существующих
  - эти правила обычно называют *аксиомами Армстронга*, хотя справедливость правил доказывается на основе определения FD
- Пусть  $A$ ,  $B$  и  $C$  являются (в общем случае, составными) атрибутами переменной отношения  $r$
- Множества  $A$ ,  $B$  и  $C$  могут иметь непустое пересечение
- Для краткости будем обозначать через  $AB$   $A \cup B$ .

# Элементы теории функциональных зависимостей (9)

## Базовые определения и утверждения (8)

- Тогда для любого значения  $r$ :
  - если  $B \subseteq A$ , то выполняется  $FD A \rightarrow B$  (аксиома рефлексивности);
  - если выполняется  $FD A \rightarrow B$ , то выполняется и  $FD AC \rightarrow BC$  (аксиома пополнения);
  - если выполняются  $FD A \rightarrow B$  и  $B \rightarrow C$ , то выполняется и  $FD A \rightarrow C$  (аксиома транзитивности)

# Элементы теории функциональных зависимостей (10)

## Базовые определения и утверждения (9)

- Докажем истинность этих «аксиом».
- Истинность первой аксиомы Армстронга ( $B \subseteq A$  влечет  $A \rightarrow B$ ) следует из того, что при  $B \subseteq A$   $FD A \rightarrow B$  является тривиальной.
- Справедливость второй аксиомы ( $A \rightarrow B$  влечет  $FD AC \rightarrow BC$ ) докажем от противного
  - Предположим, что  $FD AC \rightarrow BC$  не соблюдается
  - Это означает, что в некотором допустимом теле значения переменной отношения  $r$  найдутся два кортежа  $t1$  и  $t2$ , такие, что  $t1 \{AC\} = t2 \{AC\}$  (a), но  $t1 \{BC\} \neq t2 \{BC\}$  (b)
    - ✓ (здесь  $t \{A\}$  обозначает проекцию кортежа  $t$  на множество атрибутов  $A$ ).
  - По аксиоме рефлексивности из равенства (a) следует, что  $t1 \{A\} = t2 \{A\}$
  - Поскольку имеется  $FD A \rightarrow B$ , должно соблюдаться равенство  $t1 \{B\} = t2 \{B\}$
  - Тогда из неравенства (b) следует, что  $t1 \{C\} \neq t2 \{C\}$ , что противоречит наличию тривиальной  $FD AC \rightarrow C$
  - Следовательно, предположение об отсутствии  $FD AC \rightarrow BC$  не является верным, и справедливость второй аксиомы доказана

# Элементы теории функциональных зависимостей (11)

## Базовые определения и утверждения (10)

- Аналогично докажем истинность третьей аксиомы Армстронга ( $A \rightarrow B$  и  $B \rightarrow C$  влечет  $A \rightarrow C$ )
  - Предположим, что FD  $A \rightarrow C$  не соблюдается
  - Это означает, что в некотором допустимом теле отношения найдутся два кортежа  $t1$  и  $t2$ , такие что  $t1 \{A\} = t2 \{A\}$ , но  $t1 \{C\} \neq t2 \{C\}$
  - Но из наличия FD  $A \rightarrow B$  следует, что  $t1 \{B\} = t2 \{B\}$ , а потому из наличия FD  $B \rightarrow C$  следует, что  $t1 \{C\} = t2 \{C\}$
  - Следовательно, предположение об отсутствии FD  $A \rightarrow C$  не является верным, и справедливость третьей аксиомы и утверждения в целом доказана

# Элементы теории функциональных зависимостей (12)

## Базовые определения и утверждения (11)

- Можно доказать, что система правил вывода Армстронга *полна и совершенна (sound and complete)* в том смысле, что
  - для данного множества FD  $S$  любая FD, потенциально выводимая из  $S$ , может быть выведена на основе аксиом Армстронга, и
  - применение этих аксиом не может привести к выводу лишней FD
- Тем не менее, Дейт по практическим соображениям предложил расширить базовый набор правил вывода еще пятью правилами,
  - для которых мы сразу покажем, что они выводятся из исходных аксиом Армстронга

# Элементы теории функциональных зависимостей (13)

## Базовые определения и утверждения (12)

- Для любого атрибута  $A$  выполняется  $FD A \rightarrow A$  (*аксиома самодетерминированности*)
  - прямо следует из аксиомы рефлексивности
- Если выполняется  $FD A \rightarrow BC$ , то выполняются и  $FD A \rightarrow B$  и  $A \rightarrow C$  (*аксиома декомпозиции*)
  - из аксиомы рефлексивности следует, что выполняется  $FD BC \rightarrow B$ ;
  - из аксиомы транзитивности следует, что выполняется  $FD A \rightarrow B$ ;
  - аналогично, из аксиомы рефлексивности следует, что выполняется  $FD BC \rightarrow C$ , а
  - из аксиомы транзитивности следует, что выполняется  $FD A \rightarrow C$

# Элементы теории функциональных зависимостей (14)

## Базовые определения и утверждения (13)

- Если выполняются  $FD A \rightarrow B$  и  $A \rightarrow C$ , то выполняется и  $FD A \rightarrow BC$  (*аксиома объединения*)
  - из аксиомы пополнения следует, что выполняется  $FD A \rightarrow AB$  и  $AB \rightarrow BC$ ;
  - из аксиомы транзитивности следует, что  $A \rightarrow BC$
- Если выполняются  $FD A \rightarrow B$  и  $C \rightarrow D$ , то выполняется и  $FD AC \rightarrow BD$  (*аксиома композиции*)
  - из аксиомы пополнения следует, что выполняются  $FD AC \rightarrow BC$  и  $BC \rightarrow BD$ ;
  - из аксиомы транзитивности (3) следует, что выполняется  $FD AC \rightarrow BD$
- Если выполняются  $FD A \rightarrow BC$  и  $B \rightarrow D$ , то выполняется и  $FD A \rightarrow BCD$  (*аксиома накопления*)
  - из аксиомы пополнения следует, что выполняется  $FD BC \rightarrow BCD$ ;
  - из аксиомы транзитивности следует, что выполняется  $FD A \rightarrow BCD$

# Элементы теории функциональных зависимостей (15)

## Базовые определения и утверждения (14)

### ■ **Определение 5.5. Замыкание множества атрибутов**

Пусть заданы переменная отношения  $r$ , множество  $Z$  атрибутов этого отношения (подмножество  $Hr$ , или составной атрибут  $r$ ) и некоторое множество FD  $S$ , выполняемых для  $r$ .

Тогда *замыканием  $Z$  над  $S$*  называется наибольшее множество  $Z^+$  таких атрибутов  $Y$  отношения  $r$ , что  $FD Z \rightarrow Y \in S^+$

# Элементы теории функциональных зависимостей (16)

## Базовые определения и утверждения (15)

### ■ Докажем корректность алгоритма по индукции

- На нулевом шаге  $Z[0] = Z$ , и FD  $Z \rightarrow Z[k]$ , очевидно, принадлежит  $S^+$  (тривиальная FD «выводится» из любого множества FD)

Алгоритм вычисления  $Z^+$

```
k := 0; Z[0] := Z;
DO
k := k+1;
Z[k] := Z[k-1];
FOR EACH FD A → B IN S DO
IF A ⊆ Z[k] THEN Z[k] := (Z[k] UNION B) END DO;
UNTIL Z[k] = Z[k-1];
Z+ := Z[k];
```

- Пусть для некоторого  $k$  выполняется FD  $Z \rightarrow Z[k]$ , и пусть мы нашли в  $S$  такую FD  $A \rightarrow B$ , что  $A \subseteq Z[k]$ .
- Тогда можно представить  $Z[k]$  в виде  $AC$ , и, следовательно, выполняется FD  $Z \rightarrow AC$ .
- Но по аксиоме накопления (8) тогда выполняется FD  $Z \rightarrow ACB$ , т.е. FD  $Z \rightarrow (Z[k] \text{ UNION } B)$  входит во множество  $S^+$ , что переводит нас на следующий шаг индукции.
- Очевидно, что, поскольку множество атрибутов отношения конечно, то на некотором шаге алгоритм завершит свою работу

# Элементы теории функциональных зависимостей (17)

## Базовые определения и утверждения (16)

- Продемонстрируем работу алгоритма на примере.
- Пусть имеется отношение с заголовком  $\{A, B, C, D, E, F\}$  и заданным множеством FD  $S = \{A \rightarrow D, AB \rightarrow E, BF \rightarrow E, CD \rightarrow F, E \rightarrow C\}$ .
- Пусть требуется найти  $\{AE\}^+$  над  $S$ .
  - На первом проходе тела цикла DO  $Z[1]$  равно  $AE$ .
  - В теле цикла FOR EACH будут найдены FD  $A \rightarrow D$  и  $E \rightarrow C$ , и в конце цикла  $Z[1]$  станет равным  $ACDE$ .
  - На втором проходе тела цикла DO при  $Z[2]$ , равном  $ACDE$ , в теле цикла FOR EACH будет найдена FD  $CD \rightarrow F$ , и в конце цикла  $Z[2]$  станет равным  $ACDEF$ .
  - Следующий проход тела цикла DO не изменит  $Z[3]$ , и  $Z^+ (\{AE\}^+)$  будет равно  $ACDEF$ .
- Алгоритм построения замыкания множества атрибутов  $Z$  над заданным множеством FD  $S$  помогает легко установить, входит ли заданная FD  $Z \rightarrow B$  в замыкание  $S^+$ .
- Очевидно, что необходимым и достаточным условием для этого является  $B \subseteq Z^+$ , т. е. входжение составного атрибута  $B$  в замыкание  $Z$

### Алгоритм вычисления $Z^+$

```
k := 0; Z[0] := Z;
DO
k := k+1;
Z[k] := Z[k-1];
FOR EACH FD A → B IN S DO
IF A ⊆ Z[k] THEN Z[k] := (Z[k] UNION B) END DO;
UNTIL Z[k] = Z[k-1];
Z+ := Z[k];
```

# Элементы теории функциональных зависимостей (18)

## Базовые определения и утверждения (17)

### ■ **Определение 5.6. Суперключ отношения**

Суперключом переменной отношения  $r$  называется любое подмножество  $K$  заголовка  $Hr$ , включающее, по меньшей мере, хотя бы один возможный ключ  $r$

- Одно из следствий этого определения состоит в том, что подмножество  $K$  заголовка  $Hr$  является суперключом тогда и только тогда, когда для любого атрибута  $A$  (возможно, составного) из заголовка отношения  $r$  выполняется  $FD K \rightarrow A$ .
- В терминах замыкания множества атрибутов  $K$  является суперключом тогда и только тогда, когда  $K^+$  совпадает с  $Hr$

# Элементы теории функциональных зависимостей (19)

## Базовые определения и утверждения (18)

### ■ **Определение 5.7. Покрытие множества FD**

Множество FD  $S_2$  называется *покрытием* множества FD  $S_1$ , если любая FD, выводимая из  $S_1$ , выводится также и из  $S_2$

- Легко заметить, что  $S_2$  является покрытием  $S_1$  тогда и только тогда, когда  $S_1^+ \subseteq S_2^+$
- Два множества FD  $S_1$  и  $S_2$  называются *эквивалентными*, если каждое из них является покрытием другого, т. е.  $S_1^+ = S_2^+$

# Элементы теории функциональных зависимостей (20)

## Базовые определения и утверждения (21)

### ■ **Определение 5.8. Минимальное множество FD**

Множество FD  $S$  называется *минимальным* в том и только в том случае, когда удовлетворяет следующим свойствам:

- правая часть любой FD из  $S$  является множеством из одного атрибута (простым атрибутом);
- детерминант каждой FD из  $S$  обладает свойством *минимальности*, т.е.
  - ✓ удаление любого атрибута из детерминанта приводит к изменению замыкания  $S^+$ , т. е. порождению множества FD, не эквивалентного  $S$
  - ✓ FD с минимальным детерминантом называется *минимальной слева*;
- удаление любой FD из  $S$  приводит к изменению  $S^+$ , т. е. порождению множества FD, не эквивалентного  $S$

# Элементы теории функциональных зависимостей (21)

## Базовые определения и утверждения (22)

- Если считать, что единственным возможным ключом отношения СЛУЖАЩИЕ\_ПРОЕКТЫ является атрибут СЛУ\_НОМ, то множество FD

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
4434	Иванов	22400.00	1	Иванов
4435	Петров	29000.00	1	Иванов
4415	Сидоров	23000.00	1	Иванов
4436	Федоров	20600.00	1	Иванов
4440	Иванова	22000.00	1	Иванов
4441	Сидоренко	18000.00	2	Иваненко
4416	Федоренко	20400.00	2	Иваненко
4417	Иваненко	21600.00	2	Иваненко

{СЛУ\_НОМ → СЛУ\_ИМЯ, СЛУ\_НОМ → СЛУ\_ЗАРП, СЛУ\_НОМ → ПРО\_НОМ, ПРО\_НОМ → ПРОЕКТ\_РУК} будет минимальным

- в правых частях FD этого множества находятся множества, состоящие ровно из одного атрибута;
- каждый из детерминантов тоже является множеством из одного атрибута, удаление которого, очевидно, недопустимо;
- удаление каждой FD явно приводит к изменению замыкания множества FD, поскольку утрачиваемая информация не выводится с помощью аксиом Армстронга

## Элементы теории функциональных зависимостей (22)

### Базовые определения и утверждения (21)

- С другой стороны, множества FD
  - {СЛУ\_НОМ → {СЛУ\_ИМЯ, СЛУ\_ЗАРП}, СЛУ\_НОМ → ПРО\_НОМ, СЛУ\_НОМ → ПРОЕКТ\_РУК, ПРО\_НОМ → ПРОЕКТ\_РУК};
  - {СЛУ\_НОМ → СЛУ\_ИМЯ, {СЛУ\_НОМ, СЛУ\_ИМЯ} → СЛУ\_ЗАРП, СЛУ\_НОМ → ПРО\_НОМ, СЛУ\_НОМ → ПРОЕКТ\_РУК, ПРО\_НОМ → ПРОЕКТ\_РУК};
  - {СЛУ\_НОМ → СЛУ\_НОМ, СЛУ\_НОМ → СЛУ\_ИМЯ, СЛУ\_НОМ → СЛУ\_ЗАРП, СЛУ\_НОМ → ПРО\_НОМ, СЛУ\_НОМ → ПРОЕКТ\_РУК, ПРО\_НОМ → ПРОЕКТ\_РУК}не являются минимальными

## Элементы теории функциональных зависимостей (23)

### Базовые определения и утверждения (22)

- Для первого множества в правой части первой FD присутствует множество из двух элементов
- Для второго множества удаление атрибута СЛУ\_ИМЯ из детерминанта FD {СЛУ\_НОМ, СЛУ\_ИМЯ}  $\rightarrow$  СЛУ\_ЗАРП не меняет замыкание множества FD
- Для третьего множества удаление FD СЛУ\_НОМ  $\rightarrow$  СЛУ\_НОМ не приводит к изменению замыкания
- Эти примеры показывают, что для определения минимальности множества FD не всегда требуется явное построение замыкания данного множества

# Элементы теории функциональных зависимостей (24)

## Базовые определения и утверждения (23)

- Для любого множества FD  $S$  существует (и даже может быть построено) эквивалентное ему минимальное множество  $S$ -
- Приведем общую схему построения  $S$ - по заданному множеству FD  $S$ 
  - Используя аксиому декомпозиции, мы можем привести множество  $S$  к эквивалентному множеству FD  $S_1$ , правые части FD которого содержат только одноэлементные множества (простые атрибуты)
  - Для каждой FD из  $S_1$ , детерминант  $L \{L_1, L_2, \dots, L_n\}$  которой содержит более одного атрибута, будем пытаться удалять атрибуты  $L_i$  ( $i = 1, 2, \dots, n$ ), получая множество FD  $S_2$
  - Если после удаления атрибута  $L_i$  множество  $S_2$  оказывается эквивалентным множеству  $S_1$ , то этот атрибут удаляется, и пробуются следующий атрибут
  - Назовем  $S_3$  множество FD, полученное путем допустимого удаления атрибутов из всех детерминантов FD множества  $S_1$
  - Для каждой FD  $f$  из множества  $S_3$  будем проверять эквивалентность множеств  $S_3$  и  $S_3 \text{ MINUS } \{f\}$
  - Если эти множества оказываются эквивалентными, удалим  $f$  из множества  $S_3$ , и в заключение получим множество  $S_4$ , которое минимально и по построению эквивалентно исходному множеству FD  $S$

# Элементы теории функциональных зависимостей (25)

## Базовые определения и утверждения (24)

- Пусть, например, имеется переменная отношения  $r$  с заголовком  $\{A, B, C, D\}$ , и задано множество FD  $S = \{A \rightarrow B, A \rightarrow BC, AB \rightarrow C, AC \rightarrow D, B \rightarrow C\}$
- По аксиоме декомпозиции  $S$  эквивалентно множеству  $\{A \rightarrow B, A \rightarrow C, AB \rightarrow C, AC \rightarrow D, B \rightarrow C\}$
- В детерминанте FD  $AC \rightarrow D$  можно удалить атрибут  $C$ , поскольку
  - по аксиоме пополнения из наличия FD  $A \rightarrow C$  следует наличие FD  $A \rightarrow AC$ ;
  - по аксиоме транзитивности выводится FD  $A \rightarrow D$ , и поэтому
  - атрибут  $C$  в детерминанте FD  $AC \rightarrow D$  является избыточным
- FD  $AB \rightarrow C$  может быть удалена, поскольку может быть выведена из FD  $A \rightarrow C$ 
  - по аксиоме пополнения из этой FD выводится FD  $AB \rightarrow BC$ ,
  - а по аксиоме декомпозиции далее выводится  $AB \rightarrow C$
- Наконец, FD  $A \rightarrow C$  тоже выводится по аксиоме транзитивности из FD  $A \rightarrow B$  и  $B \rightarrow C$
- Таким образом, мы получаем множество FD  $\{A \rightarrow B, A \rightarrow D, B \rightarrow C\}$ , которое является минимальным и эквивалентно  $S$  по построению

# Элементы теории функциональных зависимостей (26)

## Базовые определения и утверждения (25)

### ■ **Определение 5.9. Минимальное покрытие множества FD**

Минимальным покрытием множества FD  $S$  называется любое минимальное множество FD  $S_1$ , эквивалентное  $S$

- Поскольку для каждого множества FD существует эквивалентное минимальное множество FD,
  - у каждого множества FD имеется хотя бы одно минимальное покрытие,
  - причем для его нахождения не обязательно находить замыкание исходного множества

# Элементы теории функциональных зависимостей (27)

## Декомпозиция без потерь и функциональные зависимости (1)

- Далее мы будем обсуждать подход к проектированию реляционных баз данных на основе нормализации,
  - т. е. декомпозиции (разбиения путем проецирования) отношения, находящегося в предыдущей нормальной форме, на два или более отношений, удовлетворяющих требованиям следующей нормальной формы
- Считаются правильными такие декомпозиции отношения, которые являются обратимыми,
  - т. е. имеется возможность собрать исходное отношение из декомпозированных отношений без потери информации
- Такие декомпозиции называются *декомпозициями без потерь*

# Элементы теории функциональных зависимостей (28)

## Декомпозиция без потерь и функциональные зависимости (2)

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
4434	Иванов	22400.00	1	Иванов
4417	Иваненко	22400.00	2	Иваненко

- Две возможные декомпозиции отношения СЛУЖАЩИЕ\_ПРОЕКТЫ

Декомпозиция (1). Отношения СЛУЖ и СЛУ\_ПРО

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП
4434	Иванов	22400.00
4417	Иваненко	22400.00

СЛУ_НОМ	ПРО_НОМ	ПРОЕКТ_РУК
4434	1	Иванов
4417	2	Иваненко

Декомпозиция (2). Отношения СЛУЖ и ЗАРП\_ПРО

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП
4434	Иванов	22400.00
4417	Иваненко	22400.00

СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
22400.00	1	Иванов
22400.00	2	Иваненко

# Элементы теории функциональных зависимостей (29)

## Декомпозиция без потерь и функциональные зависимости (3)

- В случае декомпозиции (1) мы не потеряли информацию о служащих
  - про каждого из них можно узнать имя, размер зарплаты, номер выполняемого проекта и имя руководителя проекта
- Вторая декомпозиция не дает возможности получить данные о проекте служащего, поскольку Иванов и Иваненко получают одинаковую зарплату
  - следовательно, эта декомпозиция приводит к потере информации
- Что же привело к тому, что одна декомпозиция является декомпозицией без потерь, а вторая – нет?

# Элементы теории функциональных зависимостей (30)

## Декомпозиция без потерь и функциональные зависимости (4)

- При выполнении декомпозиции мы использовали операцию взятия проекции
  - Каждое из значений отношений СЛУЖ, СЛУ\_ПРО и ЗАРП\_ПРО является проекцией исходного значения отношения СЛУЖАЩИЕ\_ПРОЕКТЫ
- В случае декомпозиции (1) отсутствие потери информации означает, что в результате естественного соединения значений отношений СЛУЖ и СЛУ\_ПРО будет гарантированно получено значение отношения,
  - заголовков и тело которого совпадают с заголовком и телом значения отношения СЛУЖАЩИЕ\_ПРОЕКТЫ
- Это произойдет для любых допустимых (и согласованных) значений переменных отношений СЛУЖАЩИЕ\_ПРОЕКТЫ, СЛУЖ и СЛУ\_ПРО, поскольку у всех этих переменных атрибут СЛУ\_НОМ является возможным ключом

# Элементы теории функциональных зависимостей (31)

## Декомпозиция без потерь и функциональные зависимости (5)

- Однако если выполнить естественное соединение значений отношений СЛУ и ЗАРП\_ПРО, то будет получено следующее значение отношения:

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
4434	Иванов	22400.00	1	Иванов
4417	Иваненко	22400.00	2	Иваненко
4434	Иванов	22400.00	2	Иваненко
4417	Иваненко	22400.00	1	Иванов

- Схема этого отношения, естественно (поскольку соединение – естественное), совпадает со схемой отношения СЛУЖАЩИЕ\_ПРОЕКТЫ, но в теле появились лишние кортежи, наличие которых и приводит к утрате исходной информации.
- Интуитивно понятно, что это происходит потому, что в отношении ЗАРП\_ПРО отсутствуют функциональные зависимости СЛУ\_ЗАРП → ПРО\_НОМ и СЛУ\_ЗАРП → ПРОЕКТ\_РУК, но точнее причину потери информации в данном случае мы объясним несколько позже
- Корректность же декомпозиции 1 следует из теоремы Хита

# Элементы теории функциональных зависимостей (32)

## Декомпозиция без потерь и функциональные зависимости (6)

### ■ Теорема Хита

Пусть задана переменная отношения  $r$  с заголовком  $\{A, B, C\}$

- $(A, B$  и  $C$ , в общем случае, являются составными атрибутами),  
и выполняется FD  $A \rightarrow B$

Тогда  $r = (r \text{ PROJECT } \{A, B\}) \text{ NATURAL JOIN } (r \text{ PROJECT } \{A, C\})$

### ■ Доказательство

- Пусть  $R$  – некоторое произвольное значение переменной  $r$
- Обозначим результат операции  $R \text{ PROJECT } \{A, B\}$  через  $R_1$ ,  
результат операции  $R \text{ PROJECT } \{A, C\}$  через  $R_2$ ,  
а результат  $R_1 \text{ NATURAL JOIN } R_2$  через  $R_3$
- Докажем, что в  $B_{R_3}$  содержатся все кортежи, содержащиеся в  $B_R$ 
  - ✓ Действительно, пусть кортеж  $\{ \langle A, v_A \rangle, \langle B, v_B \rangle, \langle C, v_C \rangle \} \in B_R$
  - ✓ Тогда по определению операции взятия проекции  $\{ \langle A, v_A \rangle, \langle B, v_B \rangle \} \in B_{R_1}$   
и  $\{ \langle A, v_A \rangle, \langle C, v_C \rangle \} \in B_{R_2}$
  - ✓ Следовательно,  $\{ \langle A, v_A \rangle, \langle B, v_B \rangle, \langle C, v_C \rangle \} \in B_{R_3}$

# Элементы теории функциональных зависимостей (33)

## Декомпозиция без потерь и функциональные зависимости (7)

- Теперь докажем, что в теле результата естественного соединения нет лишних кортежей, т. е. что
  - ✓ если кортеж  $\{\langle A, v_A \rangle, \langle B, v_B \rangle, \langle C, v_C \rangle\} \in B_{R3}$ ,
  - ✓ то  $\{\langle A, v_A \rangle, \langle B, v_B \rangle, \langle C, v_C \rangle\} \in B_R$
- ✓ Действительно, если  $\{\langle A, v_A \rangle, \langle B, v_B \rangle, \langle C, v_C \rangle\} \in B_{R3}$ , то существуют кортежи  $\{\langle A, v_A \rangle, \langle B, v_B \rangle\} \in B_{R1}$  и  $\{\langle A, v_A \rangle, \langle C, v_C \rangle\} \in B_{R2}$
- ✓ Последнее условие может выполняться в том и только в том случае, когда существует кортеж  $\{\langle A, v_A \rangle, \langle B, v_B^* \rangle, \langle C, v_C \rangle\} \in B_R$
- ✓ Но поскольку выполняется FD  $A \rightarrow B$ , то  $v_B = v_B^*$  и, следовательно,  $\{\langle A, v_A \rangle, \langle B, v_B \rangle, \langle C, v_C \rangle\} = \{\langle A, v_A \rangle, \langle B, v_B^* \rangle, \langle C, v_C \rangle\}$

# Элементы теории функциональных зависимостей (34)

## Декомпозиция без потерь и функциональные зависимости (8)

- Иллюстрация общего случая применения теоремы Хита
- Атрибут СЛУ\_ОТД содержит номера отделов, в которых работают служащие, а ПРО\_НОМ – номера проектов, в которых служащие принимают участие
- Каждый служащий работает только в одном отделе, т. е. имеется FD СЛУ\_НОМ → СЛУ\_ОТД, но один служащий может участвовать в нескольких проектах
- Атрибут СЛУ\_НОМ не является возможным ключом, но наличия FD СЛУ\_НОМ\_СЛУ\_ОТД оказывается достаточно для декомпозиции этого отношения без потерь

СЛУЖАЩИЕ\_ОТДЕЛЫ\_ПРОЕКТЫ

СЛУ_НОМ	СЛУ_ОТД	ПРО_НОМ
4434	625	1
4434	625	2
4417	636	1
4417	636	2

СЛУЖ\_ОТДЕЛЫ и СЛУЖ\_ПРОЕКТЫ

СЛУ_НОМ	СЛУ_ОТД
4434	625
4417	636

СЛУ_НОМ	ПРО_НОМ
4434	1
4434	2
4417	1
4417	2

# Элементы теории функциональных зависимостей (35)

## Декомпозиция без потерь и функциональные зависимости (9)

### ■ **Определение 5.10. Минимально зависимые атрибуты**

Атрибут  $B$  минимально зависит от атрибута  $A$ , если выполняется минимальная слева  $FD A \rightarrow B$

- Например, в отношении СЛУЖАЩИЕ\_ПРОЕКТЫ выполняются  $FD$  СЛУ\_НОМ  $\rightarrow$  СЛУ\_ЗАРП и  $\{СЛУ\_НОМ, СЛУ\_ИМЯ\} \rightarrow СЛУ\_ЗАРП$

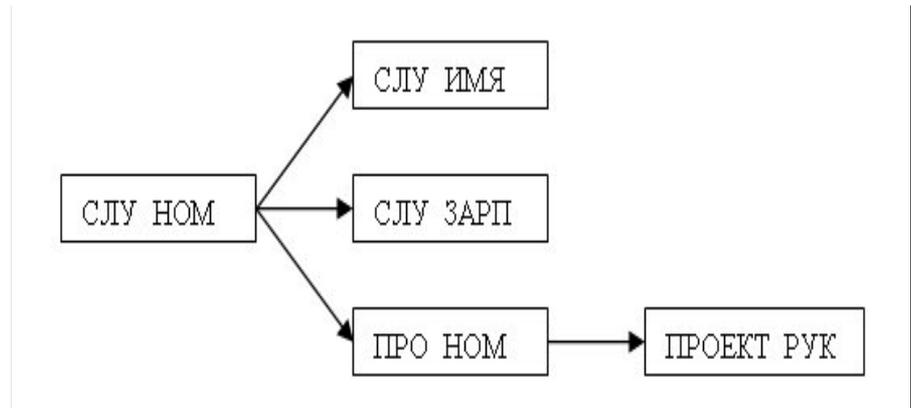
СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
4434	Иванов	22400.00	1	Иванов
4435	Петров	29000.00	1	Иванов
4415	Сидоров	23000.00	1	Иванов
4436	Федоров	20600.00	1	Иванов
4440	Иванова	22000.00	1	Иванов
4441	Сидоренко	18000.00	2	Иваненко
4416	Федоренко	20400.00	2	Иваненко
4417	Иваненко	21600.00	2	Иваненко

- Первая  $FD$  является минимальной слева, а вторая – нет
- Поэтому СЛУ\_ЗАРП минимально зависит от СЛУ\_НОМ, а для  $\{СЛУ\_НОМ, СЛУ\_ИМЯ\}$  свойство минимальной зависимости не выполняется

# Элементы теории функциональных зависимостей (36)

## Декомпозиция без потерь и функциональные зависимости (10)

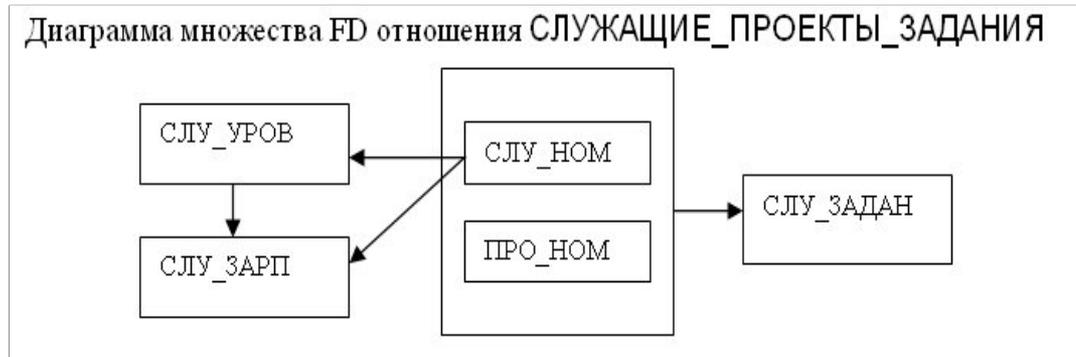
- Диаграмма минимального множества FD отношения СЛУЖАЩИЕ\_ПРОЕКТЫ
- В левой части диаграммы все стрелки начинаются с атрибута СЛУ\_НОМ, который является единственным возможным (и, следовательно, первичным) ключом отношения СЛУЖАЩИЕ\_ПРОЕКТЫ
- Отсутствует стрелка от СЛУ\_НОМ к ПРОЕКТ\_РУК
- Конечно, поскольку СЛУ\_НОМ является возможным ключом, должна выполняться и FD СЛУ\_НОМ → ПРОЕКТ\_РУК
- Но эта FD является транзитивной (через ПРО\_НОМ) и поэтому не входит в минимальное множество FD



# Минимальные FD и вторая нормальная форма (1)

- Пусть имеется переменная отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ с заголовком {СЛУ\_НОМ, СЛУ\_УРОВ, СЛУ\_ЗАРП, ПРО\_НОМ, СЛУ\_ЗАДАН}
- Атрибуты СЛУ\_УРОВ и СЛУ\_ЗАДАН содержат, соответственно, данные о разряде служащего и о задании, которое выполняет служащий в данном проекте
- Будем считать,
  - что разряд служащего определяет размер его заработной платы, и
  - что каждый служащий может участвовать в нескольких проектах, но в каждом проекте выполняет только одно задание
- Тогда очевидно, что единственно возможным ключом отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ является составной атрибут {СЛУ\_НОМ, ПРО\_НОМ}

# Минимальные FD и вторая нормальная форма (2)



Возможное значение переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ

СЛУ_НОМ	СЛУ_УРОВ	СЛУ_ЗАРП	ПРО_НОМ	СЛУ_ЗАДАН
4434	2	22400.00	1	A
4435	3	29600.00	1	B
4415	1	20000.00	1	C
4436	1	20000.00	1	D
4434	2	22400.00	2	D
4435	3	29600.00	2	C
4415	1	20000.00	2	B
4436	1	20000.00	2	A

# Минимальные FD и вторая нормальная форма (3)

## Аномалии обновления из-за наличия не минимальных FD (1)

- Во множество FD отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ входит несколько FD, в которых детерминантом является не возможный ключ отношения
  - соответствующие стрелки в диаграмме начинаются не с {СЛУ\_НОМ, ПРО\_НОМ},
  - т. е. некоторые функциональные зависимости атрибутов от возможного ключа не являются минимальными
- Это приводит к проявлению так называемых *аномалий обновления*
- Под аномалиями обновления понимаются трудности, с которыми приходится сталкиваться при выполнении операций
  - добавления кортежей в отношение (INSERT),
  - удаления кортежей (DELETE) и
  - модификации кортежей (UPDATE)

# Минимальные FD и вторая нормальная форма (4)

## Аномалии обновления из-за наличия не минимальных FD (2)

- Обсудим сначала аномалии обновления, вызываемые наличием FD СЛУ\_НОМ → СЛУ\_УРОВ
- Эти аномалии связаны с избыточностью хранения значений атрибутов СЛУ\_УРОВ и СЛУ\_ЗАРП в каждом кортеже, описывающем задание служащего в некотором проекте
- *Добавление кортежей*
- Невозможно дополнить отношение СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ данными о служащем, который в данное время еще не участвует ни в одном проекте
  - ПРО\_НОМ является частью первичного ключа и не может содержать неопределенных значений
- Между тем часто бывает, что сначала служащего принимают на работу, устанавливают его разряд и размер зарплаты, а лишь потом назначают для него проект

# Минимальные FD и вторая нормальная форма (5)

## Аномалии обновления из-за наличия не минимальных FD (3)

- *Удаление кортежей*
- Невозможно сохранить в отношении СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ данные о служащем, завершившем участие в своем последнем проекте
  - по той причине, что значение атрибута ПРО\_НОМ для этого служащего стало бы неопределенным
- Между тем характерна ситуация, когда между проектами возникают перерывы, не приводящие к увольнению служащих.
- *Модификация кортежей*
  - Чтобы изменить разряд служащего, придется модифицировать все кортежи с соответствующим значением атрибута СЛУ\_НОМ
  - В противном случае будет нарушена естественная FD СЛУ\_НОМ → СЛУ\_УРОВ
    - ✓ у одного служащего имеется только один разряд

# Минимальные FD и вторая нормальная форма (6)

## Возможная декомпозиция (1)



На основании теоремы Хита эта декомпозиция является декомпозицией без потерь, поскольку в исходном отношении имелась FD  $\{СЛУ\_НОМ, ПРО\_НОМ\} \rightarrow СЛУ\_ЗАДАН$

Значения переменных отношений СЛУЖ и СЛУЖ\_ПРО\_ЗАДАН

СЛУЖ

СЛУ_НОМ	СЛУ_УРОВ	СЛУ_ЗАРП
4434	2	22400.00
4435	3	29600.00
4415	1	20000.00
4436	1	20000.00

СЛУЖ\_ПРО\_ЗАДАН

СЛУ_НОМ	ПРО_НОМ	СЛУ_ЗАДАН
4434	1	A
4435	1	B
4415	1	C
4436	1	D
4434	2	D
4435	2	C
4415	2	B
4436	2	A

# Минимальные FD и вторая нормальная форма (7)

## Возможная декомпозиция (2)

- Теперь можно легко справиться с операциями обновления
- *Добавление кортежей*
  - Чтобы сохранить данные о принятом на работу служащем, который еще не участвует ни в каком проекте, достаточно добавить соответствующий кортеж в отношение СЛУЖ
- *Удаление кортежей*
  - Если некоторый служащий прекращает работу в некотором проекте, достаточно удалить соответствующий кортеж из отношения СЛУЖ\_ПРО\_ЗАДАН
  - При увольнении служащего нужно удалить кортежи с соответствующим значением атрибута СЛУ\_НОМ из отношений СЛУЖ и СЛУЖ\_ПРО\_ЗАДАН
- *Модификация кортежей*
  - Если у служащего меняется разряд (и, следовательно, размер зарплаты), достаточно модифицировать один кортеж в отношении СЛУЖ

# Минимальные FD и вторая нормальная форма (8)

## Вторая нормальная форма (1)

- В отношениях СЛУЖ и СЛУЖ\_ПРО\_ЗАДАН отсутствуют FD, не являющиеся минимальными
- Наличие таких FD в переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ вызывало аномалии обновления
- В действительности, проблема заключалась в том, что атрибут СЛУЖ\_УРОВ относился к сущности служащий, в то время как первичный ключ идентифицировал сущность задание\_служащего\_в\_проекте

### ■ **Определение 5.11. Вторая нормальная форма**

Переменная отношения находится во второй нормальной форме (2NF) тогда и только тогда, когда она находится в первой нормальной форме, и каждый ее неключевой атрибут минимально функционально зависит от первичного ключа

- Неключевым атрибутом называется атрибут, не входящий ни в один возможный ключ

# Минимальные FD и вторая нормальная форма (9)

## Вторая нормальная форма (2)

- Переменные отношений СЛУЖ и СЛУЖ\_ПРО\_ЗАДАН находятся в 2NF
  - все неключевые атрибуты отношений минимально зависят от первичных ключей СЛУ\_НОМ и {СЛУ\_НОМ, ПРО\_НОМ} соответственно
- Переменная отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ не находится в 2NF
  - например, FD {СЛУ\_НОМ, ПРО\_НОМ} → СЛУ\_УРОВ не является минимальной
- Любая переменная отношения, находящаяся в 1NF, может быть приведена к набору из переменных отношений, находящихся в 2NF
- В результате декомпозиции мы получаем набор проекций исходной переменной отношения, естественное соединение любых значений которых воспроизводит допустимое значение исходной переменной отношения
  - т. е. это декомпозиция без потерь
- Для переменных отношений СЛУЖ и СЛУЖ\_ПРО\_ЗАДАН исходное отношение СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ воспроизводится их естественным соединением по общему атрибуту СЛУ\_НОМ

## Минимальные FD и вторая нормальная форма (10)

### Вторая нормальная форма (3)

- Заметим, что допустимое значение переменной отношения СЛУЖ может содержать кортежи, информационное наполнение которых выходит за пределы допустимых значений переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ
- Например, в теле отношения СЛУЖ может находиться кортеж с данными о служащем с номером 4438, который еще не участвует ни в одном проекте
- Наличие такого кортежа не влияет на результат естественного соединения, тело которого все равно будет совпадать с телом допустимого значения переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ

# Нетранзитивные FD и 3NF (1)



- В произведенной декомпозиции переменной отношения СЛУЖАЩИЕ\_ПРОЕКТЫ\_ЗАДАНИЯ множество FD переменной отношения СЛУЖ\_ПРО\_ЗАДАН предельно просто – детерминантом единственной нетривиальной функциональной зависимости является возможный ключ
- При использовании этой переменной отношения какие-либо аномалии обновления не возникают
- Однако переменная отношения СЛУЖ не является такой же совершенной

# Нетранзитивные FD и 3NF (2)

## Аномалии обновления из-за наличия транзитивных FD (1)



- Функциональные зависимости переменной отношения СЛУЖ по-прежнему порождают некоторые аномалии обновления
- Они вызываются наличием транзитивной FD  $СЛУ\_НОМ \rightarrow СЛУ\_ЗАРП$ 
  - через FD  $СЛУ\_НОМ \rightarrow СЛУ\_УРОВ$  и  $СЛУ\_УРОВ \rightarrow СЛУ\_ЗАРП$
- Эти аномалии связаны с избыточностью хранения значения атрибута СЛУ\_ЗАРП в каждом кортеже, характеризующем служащих с одним и тем же разрядом

# Нетранзитивные FD и 3NF (3)

## Аномалии обновления из-за наличия транзитивных FD (2)

- *Добавление кортежей*
  - Невозможно сохранить данные о новом разряде (и соответствующем ему размере зарплаты), пока не появится служащий с новым разрядом
    - ✓ Атрибут СЛУ\_НОМ, входящий в состав первичного ключа, не может содержать неопределенные значения
- *Удаление кортежей*
  - При увольнении последнего служащего с данным разрядом будет утрачена информация о наличии такого разряда и соответствующем размере зарплаты
- *Модификация кортежей*
  - При изменении размера зарплаты, соответствующей некоторому разряду, придется изменить значение атрибута СЛУ\_ЗАРП в кортежах всех служащих, которым назначен этот разряд
    - ✓ иначе не будет выполняться FD СЛУ\_УРОВ → СЛУ\_ЗАРП

# Нетранзитивные FD и 3NF (4)

## Возможная декомпозиция (1)

Диаграммы FD в отношениях СЛУЖ1 и УРОВ



- По теореме Хита, это снова декомпозиция без потерь по причине наличия, например, FD СЛУ\_НОМ → СЛУ\_УРОВ
- Мы избавились от трудностей при выполнении операций обновления

Тела отношений СЛУЖ1 и УРОВ

СЛУЖ1

СЛУ_НОМ	СЛУ_УРОВ
4434	2
4435	3
4415	1
4436	1

УРОВ

СЛУ_УРОВ	СЛУ_ЗАРП
2	22400.00
3	29600.00
1	20000.00

# Нетранзитивные FD и 3NF (5)

## Возможная декомпозиция (2)

- *Добавление кортежей*
  - Чтобы сохранить данные о новом разряде, достаточно добавить соответствующий кортеж к отношению УРОВ
- *Удаление кортежей*
  - При увольнении последнего служащего, обладающего данным разрядом, удаляется соответствующий кортеж из отношения СЛУЖ1, но данные о разряде сохраняются в отношении УРОВ
- *Модификация кортежей*
  - При изменении размера зарплаты, соответствующей некоторому разряду, изменяется значение атрибута СЛУ\_ЗАРП ровно в одном кортеже отношения УРОВ

# Нетранзитивные FD и 3NF (6)

## Третья нормальная форма (1)

- Аномалии обновления переменной отношения СЛУЖ были связаны с наличием в этой переменной транзитивной FD СЛУ\_НОМ  $\rightarrow$  СЛУ\_ЗАРП
- Наличие этой FD на самом деле означало, что атрибут СЛУ\_ЗАРП характеризовал не сущность *служащий*, а сущность *разряд*
- **Определение 5.12. Третья нормальная форма**

Переменная отношения находится в третьей нормальной форме (3NF) в том и только в том случае, когда она находится во второй нормальной форме, и каждый неключевой атрибут нетранзитивно функционально зависит от первичного ключа

- Функциональная зависимость называется нетранзитивной тогда и только тогда, когда она не является транзитивной

# Нетранзитивные FD и 3NF (7)

## Третья нормальная форма (2)

- Отношения СЛУЖ1 и УРОВ оба находятся в 3NF
  - все неключевые атрибуты нетранзитивно зависят от первичных ключей СЛУ\_НОМ и СЛУ\_УРОВ
- Отношение СЛУЖ не находится в 3NF
  - FD СЛУ\_НОМ → СЛУ\_ЗАРП является транзитивной
- Любое отношение, находящееся в 2NF, но не находящееся в 3NF, может быть приведено к набору отношений, находящихся в 3NF
- При этом получается набор проекций исходного отношения, естественное соединение которых воспроизводит исходное отношение
  - т. е. это декомпозиция без потерь
- Для отношений СЛУЖ1 и УРОВ исходное отношение СЛУЖ воспроизводится их естественным соединением по общему атрибуту СЛУ\_УРОВ

## Нетранзитивные FD и 3NF (8)

### Третья нормальная форма (3)

- Заметим, что допустимые значения отношения УРОВ могут содержать кортежи, информационное наполнение которых выходит за пределы тела отношения СЛУЖ
- Например, в теле отношения УРОВ может находиться кортеж с данными о разряде 4, который еще не присвоен ни одному служащему
- Наличие такого кортежа не влияет на результат естественного соединения, который все равно будет являться допустимым значением отношения СЛУЖ

# Нетранзитивные FD и 3NF (9)

## Независимые проекции отношений. Теорема Риссанена (1)

- Заметим, что для переменной отношения СЛУЖ с заголовком {СЛУ\_НОМ, СЛУ\_УРОВ, СЛУ\_ЗАРП}, кроме декомпозиции на переменные отношений СЛУЖ1 с заголовком {СЛУ\_НОМ, СЛУ\_УРОВ} и УРОВ с заголовком {СЛУ\_УРОВ, СЛУ\_ЗАРП}, возможна и декомпозиция на отношения СЛУЖ1 с заголовком {СЛУ\_НОМ, СЛУ\_УРОВ} и СЛУЖ\_ЗАРП с заголовком {СЛУ\_НОМ, СЛУ\_ЗАРП}
  - теоретически возможная третья декомпозиция отношения СЛУЖ на отношения СЛУЖ2 с заголовком {СЛУ\_НОМ, СЛУ\_ЗАРП} и УРОВ с заголовком {СЛУ\_УРОВ, СЛУ\_ЗАРП} не является декомпозицией без потерь
- Обе переменные отношения, полученные путем второй декомпозиции, находятся в 3NF, и эта декомпозиция также является декомпозицией без потерь
- Тем не менее, вторая декомпозиция, в отличие от первой, не устраняет проблемы, связанные с обновлением отношения СЛУЖ
- Например, по-прежнему невозможно сохранить данные о разряде, которым не обладает ни один служащий
- Посмотрим, с чем это связано

# Нетранзитивные FD и 3NF (10)

## Независимые проекции отношений. Теорема Риссанена (2)

- Переменные отношений СЛУЖ1 и УРОВ могут обновляться независимо (являются *независимыми проекциями*), и при этом результат их естественного соединения всегда будет таким, как если бы обновлялось исходная переменная отношения СЛУЖ
- Это происходит потому, что функциональные зависимости отношения СЛУЖ трансформировались в индивидуальные ограничения первичного ключа отношений СЛУЖ1 и УРОВ
- При второй декомпозиции FD СЛУ\_УРОВ  $\rightarrow$  СЛУ\_ЗАРП трансформируется в ограничение целостности сразу для двух отношений
  - такого рода ограничения целостности называются ограничениями базы данных, и их поддержка гораздо более накладна с технической точки зрения
- Понятно, что в процессе нормализации декомпозиция отношения на независимые проекции является предпочтительной
- Необходимые и достаточные условия независимости проекций отношения обеспечивает теорема Риссанена

# Нетранзитивные FD и 3NF (10)

## Независимые проекции отношений. Теорема Риссанена (2)

### ■ Теорема Риссанена

Проекции  $r_1$  и  $r_2$  переменной отношения  $r$  являются независимыми тогда и только тогда, когда:

- каждая FD в отношении  $r$  логически следует (т.е. выводится на основе аксиом Армстронга) из FD в  $r_1$  и  $r_2$ ;
- общие атрибуты  $r_1$  и  $r_2$  образуют возможный ключ хотя бы для одного из этих отношений

- Мы не будем приводить доказательство этой теоремы, но продемонстрируем ее верность на примере двух упомянутых ранее декомпозиций отношения СЛУЖ

# Нетранзитивные FD и 3NF (11)

## Независимые проекции отношений. Теорема Риссанена (3)

- В первой декомпозиции (на проекции СЛУЖ1 {СЛУ\_НОМ, СЛУ\_УРОВ} и УРОВ {СЛУ\_УРОВ, СЛУ\_ЗАРП})
  - общий атрибут СЛУ\_УРОВ является возможным (и первичным) ключом отношения УРОВ, а
  - единственная дополнительная FD отношения СЛУЖ (СЛУ\_НОМ → СЛУ\_ЗАРП) логически следует из FD СЛУ\_НОМ → СЛУ\_УРОВ и СЛУ\_УРОВ → СЛУ\_ЗАРП, выполняемых для отношений СЛУЖ1 и УРОВ соответственно
- Вторая декомпозиция (СЛУЖ1 {СЛУ\_НОМ, СЛУ\_УРОВ} и СЛУЖ\_ЗАРП {СЛУ\_НОМ, СЛУ\_ЗАРП}) удовлетворяет второму условию теоремы Риссанена
  - СЛУ\_НОМ является первичным ключом в каждом из отношений СЛУЖ1 и СЛУ\_ЗАРП, но FD СЛУ\_УРОВ → СЛУ\_ЗАРП не выводится из FD СЛУ\_НОМ → СЛУ\_УРОВ и СЛУ\_НОМ → СЛУ\_ЗАРП

# Нетранзитивные FD и 3NF (12)

## Независимые проекции отношений. Теорема Риссанена (4)

- *Атомарной* переменной отношения называется такая переменная, которую невозможно декомпозировать на независимые проекции
- Далеко не всегда для неатомарных (не являющихся атомарными) переменных отношений требуется декомпозиция на атомарные проекции
- Например, переменная отношения СЛУЖ2 с заголовком {СЛУ\_НОМ, СЛУ\_ЗАРП, ПРО\_НОМ} и множеством FD {СЛУ\_НОМ → СЛУ\_ЗАРП, СЛУ\_НОМ → ПРО\_НОМ} не является атомарной
  - возможна декомпозиция на независимые проекции СЛУЖ3 с заголовком {СЛУ\_НОМ, СЛУ\_ЗАРП} и СЛУЖ4 с заголовком {СЛУ\_НОМ, ПРО\_НОМ}
- Но эта декомпозиция не улучшает свойства переменной отношения СЛУЖ2 и поэтому не является осмысленной
- Другими словами, при выборе способа декомпозиции нужно стремиться к получению независимых, но не обязательно атомарных проекций

## Перекрывающиеся возможные ключи и BCNF (1)

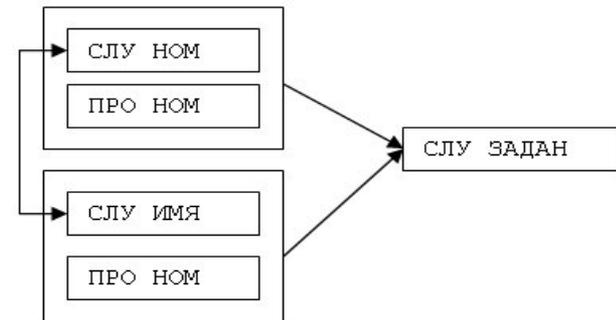
- До сих пор в определениях нормальных формы предполагали, что у декомпозируемого отношения имеется только один возможный ключ
- На практике чаще всего бывает именно так
- Но имеется один частный случай, который почти удовлетворяет требованиям 2NF и 3NF, но, тем не менее, порождает аномалии обновления
- Это тот случай, когда у отношения имеется несколько возможных ключей, и некоторые из этих возможных ключей «перекрываются», т. е. содержат общие атрибуты

## Перекрывающиеся возможные ключи и BCNF (2)

### Аномалии обновлений из-за наличия перекрывающихся возможных ключей (1)

- В отношении СЛУЖ\_ПРО\_ЗАДАН1 служащие уникально идентифицируются как по номерам удостоверений, так и по именам
- Следовательно, существуют FD  $\text{СЛУ\_НОМ} \rightarrow \text{СЛУ\_ИМЯ}$  и  $\text{СЛУ\_ИМЯ} \rightarrow \text{СЛУ\_НОМ}$
- Но один служащий может участвовать в нескольких проектах, поэтому возможными ключами являются  $\{\text{СЛУ\_НОМ}, \text{ПРО\_НОМ}\}$  и  $\{\text{СЛУ\_ИМЯ}, \text{ПРО\_НОМ}\}$
- Все FD неключевых атрибутов от возможных ключей являются минимальными, и транзитивные FD отсутствуют, но, тем не менее, этому отношению свойственны аномалии обновления

Диаграмма FD переменной отношения СЛУЖ\_ПРО\_ЗАДАН1



Возможное значение переменной отношения СЛУЖ\_ПРО\_ЗАДАН1

СЛУ_НОМ	СЛУ_ИМЯ	ПРО_НОМ	ПРО_ЗАДАН
4434	Иванов	1	А
4417	Иваненко	2	В
4434	Иванов	2	В
4417	Иваненко	1	А

## Перекрывающиеся возможные ключи и BCNF (3)

Аномалии обновлений из-за наличия перекрывающихся возможных ключей (2)

- Например, в случае изменения имени служащего требуется обновить атрибут СЛУ\_ИМЯ во всех кортежах отношения СЛУЖ\_ПРО\_ЗАДАН1, соответствующих данному служащему
- Иначе будет нарушена FD СЛУ\_НОМ  $\rightarrow$  СЛУ\_ИМЯ, и база данных окажется в несогласованном состоянии
- Причиной аномалий является то, что в требованиях 2NF и 3NF не требовалась минимальная функциональная зависимость от первичного ключа атрибутов, являющихся компонентами других возможных ключей

# Перекрывающиеся возможные ключи и BCNF (3)

## Нормальная форма Бойса-Кодда (1)

- Проблему решает нормальная форма, которую исторически принято называть нормальной формой Бойса-Кодда, и которая является уточнением 3NF в случае наличия нескольких перекрывающихся возможных ключей

### ■ **Определение 7.3. Нормальная форма Бойса-Кодда**

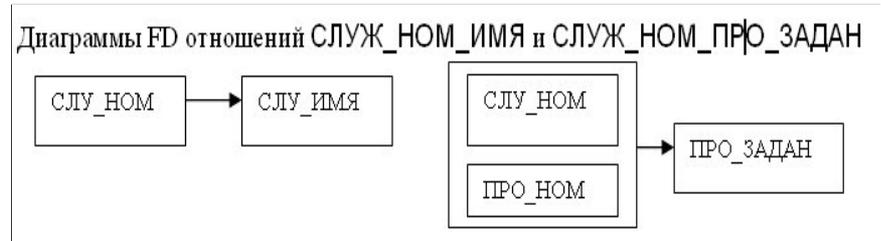
Переменная отношения находится в нормальной форме Бойса-Кодда (BCNF) в том и только в том случае, когда детерминантом любой выполняемой для этой переменной нетривиальной и минимальной FD является некоторый возможный ключ данного отношения

- Заметим, что, если в переменной отношения имеется только один возможный ключ, то любое отношение, находящееся в нормальной форме Бойса-Кодда, одновременно находится во второй и третьей нормальных формах
- Это утверждение легко доказывается методом «от противного» на основе определений 2NF и 3NF

# Перекрывающиеся возможные ключи и BCNF (4)

## Нормальная форма Бойса-Кодда (2)

- Переменная отношения СЛУЖ\_ПРО\_ЗАДАН1 может быть приведена к BCNF путем одной из двух декомпозиций:
  - СЛУЖ\_НОМ\_ИМЯ (показана на рисунке) и
  - СЛУЖ\_НОМ\_ИМЯ {СЛУ\_НОМ, СЛУ\_ИМЯ} и СЛУЖ\_ИМЯ\_ПРО\_ЗАДАН {СЛУ\_ИМЯ, ПРО\_НОМ, СЛУ\_ЗАДАН} (FD и значения отношений выглядят аналогично)
- Очевидно, что каждая из декомпозиций устраняет трудности, связанные с обновлением отношения СЛУЖ\_ПРО\_ЗАДАН1



Значения переменных отношений  
СЛУЖ\_НОМ\_ИМЯ и СЛУЖ\_НОМ\_ПРО\_ЗАДАН

СЛУЖ\_НОМ\_ИМЯ

СЛУ_НОМ	СЛУ_ИМЯ
4434	Иванов
4417	Иваненко

СЛУЖ\_ПРО\_ЗАДАН

СЛУ_НОМ	ПРО_НОМ	ПРО_ЗАДАН
4434	1	А
4417	2	В
4434	2	В
4417	1	А

# Перекрывающиеся возможные ключи и BCNF (5)

## Всегда ли следует стремиться к BCNF? (1)

- Предположим теперь, что в организации все проекты включают разные задания, и по-прежнему каждый служащий может участвовать в нескольких проектах, но может выполнять в каждом проекте только одно задание
- Одно задание в каждом проекте могут выполнять несколько служащих



Возможное значение переменной отношения  
СЛУЖ\_ПРО\_ЗАДАН (новый вариант)

СЛУ_НОМ	ПРО_НОМ	СЛУ_ЗАДАН
4434	1	A
4417	1	A
4435	2	B
4434	2	C
4417	1	D

# Перекрывающиеся возможные ключи и BCNF (6)

## Всегда ли следует стремиться к BCNF? (2)

- В переменной отношении СЛУЖ\_ПРО\_ЗАДАН существуют два возможных ключа: {СЛУ\_НОМ, ПРО\_НОМ} и {СЛУ\_НОМ, СЛУ\_ЗАДАН}
- Отношение удовлетворяет требованиям 3NF (если не учитывать то, что в нем имеются два возможных ключа):
  - отсутствуют не минимальные FD неключевых атрибутов от возможных ключей (поскольку нет неключевых атрибутов) и
  - отсутствуют транзитивные FD



Возможное значение переменной отношения СЛУЖ\_ПРО\_ЗАДАН (новый вариант)

СЛУ_НОМ	ПРО_НОМ	СЛУ_ЗАДАН
4434	1	A
4417	1	A
4435	2	B
4434	2	C
4417	1	D

# Перекрывающиеся возможные ключи и BCNF (6)

## Всегда ли следует стремиться к BCNF? (2)

- Однако из-за наличия FD  $СЛУ\_ЗАДАН \rightarrow ПРО\_НОМ$  это отношение не находится в BCNF
- Поэтому отношению  $СЛУ\_ПРО\_ЗАДАН$  снова свойственны аномалии обновления
- Например, поскольку  $СЛУ\_НОМ$  является компонентом обоих возможных ключей, невозможно удалить данные о единственном служащем, выполняющем задание в некотором проекте, не утратив информацию об этом задании



Возможное значение переменной отношения  
 $СЛУЖ\_ПРО\_ЗАДАН$  (новый вариант)

СЛУ_НОМ	ПРО_НОМ	СЛУ_ЗАДАН
4434	1	A
4417	1	A
4435	2	B
4434	2	C
4417	1	D

# Перекрывающиеся возможные ключи и BCNF (7)

## Всегда ли следует стремиться к BCNF? (3)

- Можно привести отношение СЛУЖ\_ПРО\_ЗАДАН к BCNF, выполнив его декомпозицию на отношения СЛУЖ\_НОМ\_ЗАДАН с заголовком {СЛУ\_НОМ, СЛУ\_ЗАДАН} и ПРО\_НОМ\_ЗАДАН с заголовком {СЛУ\_ЗАДАН, ПРО\_НОМ}
- Единственным возможным ключом отношения СЛУЖ\_НОМ\_ЗАДАН является {СЛУ\_НОМ, СЛУ\_ЗАДАН}, и в этом отношении отсутствуют нетривиальные FD
- Эта декомпозиция решает указанные выше проблемы
  - теперь можно хранить данные о задании проекта, не выполняемом ни одним служащим

Значения переменных отношений  
СЛУЖ\_НОМ\_ЗАДАН и ПРО\_НОМ\_ЗАДАН

СЛУЖ\_НОМ\_ЗАДАН

СЛУ_НОМ	СЛУ_ЗАДАН
4434	A
4417	A
4435	B
4434	C
4417	D

ПРО\_НОМ\_ЗАДАН

ПРО_НОМ	СЛУ_ЗАДАН
1	A
2	B
2	C
1	D

# Перекрывающиеся возможные ключи и BCNF (8)

## Всегда ли следует стремиться к BCNF? (4)

- Однако возникают новые трудности
- Например, система должна запретить добавление в отношение СЛУЖ\_НОМ\_ЗАДАН кортежа {<СЛУ\_НОМ, 4434>, <СЛУ\_ЗАДАН, D>}, поскольку задание D относится к проекту 1, а служащий с номером 2934 уже выполняет задание в этом проекте
- Исходная FD {СЛУ\_НОМ, ПРО\_НОМ} → СЛУ\_ЗАДАН не выводится из единственной (нетривиальной) действующей для этих проекций FD СЛУ\_ЗАДАН → ПРО\_НОМ, и соответствующее ограничение целостности становится ограничением базы данных

Значения переменных отношений  
СЛУЖ\_НОМ\_ЗАДАН и ПРО\_НОМ\_ЗАДАН

СЛУЖ\_НОМ\_ЗАДАН

СЛУ_НОМ	СЛУ_ЗАДАН
4434	A
4417	A
4435	B
4434	C
4417	D

ПРО\_НОМ\_ЗАДАН

ПРО_НОМ	СЛУ_ЗАДАН
1	A
2	B
2	C
1	D

## Перекрывающиеся возможные ключи и BCNF (8)

### Всегда ли следует стремиться к BCNF? (4)

- Тем самым, проекции СЛУЖ\_НОМ\_ЗАДАН и ПРО\_НОМ\_ЗАДАН не являются независимыми, а переменная отношения СЛУЖ\_ПРО\_ЗАДАН атомарна, хотя и не находится в BCNF
- Из этого следует, что при проектировании реляционной базы данных приведение отношения к BCNF не должно быть самоцелью
- Нужно внимательно оценивать положительные и отрицательные последствия нормализации

# Перекрывающиеся возможные ключи и BCNF (9)

## Всегда ли следует стремиться к BCNF? (5)

- Наконец, приведем пример, когда наличие двух перекрывающихся возможных ключей не мешает отношению находиться в BCNF
- Предположим, что в организации проекты включают одни и те же задания, каждый служащий может участвовать в нескольких проектах, но может выполнять в каждом проекте только одно задание



# Перекрывающиеся возможные ключи и BCNF (10)

## Всегда ли следует стремиться к BCNF? (6)

- В третьем варианте отношения СЛУЖ\_НОМ\_ЗАДАН имеются перекрывающиеся возможные ключи ( $\{СЛУ\_НОМ, ПРО\_НОМ\}$  и  $\{ПРО\_НОМ, СЛУ\_ЗАДАН\}$ ), однако оно находится в BCNF, поскольку эти ключи являются единственными детерминантами имеющих нетривиальных и минимальных FD
- Легко убедиться, что этому отношению аномалии обновления не свойственны



# Заключение (1)

- В первом разделе этой темы было введено понятие функциональной зависимости и исследовались важные свойства функциональных зависимостей
- Одна из целей состояла в том, чтобы на основе некоторого множества функциональных зависимостей суметь построить минимальное эквивалентное множество функциональных зависимостей
- Мы начали обсуждение с понятия замыканий множества функциональных зависимостей и аксиом Амстронга, теоретически позволяющих построить такое замыкание

# Заключение (2)

- Замыкание множества функциональных зависимостей содержит все функциональные зависимости, выводимые из функциональных зависимостей заданного множества
- Рассмотренный далее алгоритм построения замыкания множества атрибутов над заданным множеством функциональных зависимостей упрощает задачу, позволяя определить принадлежность заданной функциональной зависимости к замыканию заданного множества функциональных зависимостей без потребности в реальном построении замыкания

# Заключение (3)

- Далее мы занялись покрытиями множеств функциональных зависимостей и минимальными множествами функциональных зависимостей
- Наиболее важным результатом этого подраздела является доказательство существования и наметки алгоритма построения минимального покрытия заданного множества функциональных зависимостей – минимального множества функциональных зависимостей, эквивалентного исходному множеству

# Заключение (4)

- Наконец, был обсужден критерий декомпозиции отношений без потерь, т. е. такой способ проецирования заданной переменной отношения на две переменных отношений, при котором результат естественного соединения значений переменных-проекций в точности совпадает со значением исходной переменной отношения
- Достаточное (и очень естественное) условие декомпозиции без потерь обеспечивает теорема Хита

# Заключение (5)

- В следующих трех разделах мы обсудили три начальные нормальные формы переменных отношений:
  - вторую и
  - третью нормальные формы и
  - нормальную форму Бойса-Кодда,получаемые путем декомпозиции без потерь исходной переменной отношения на две переменные-проекции, в которых отсутствуют аномалии изменений, существовавшие в исходной переменной отношения по причине наличия функциональных зависимостей с нежелательными свойствами

# Заключение (6)

- Нормализация схемы базы данных способствует более эффективному выполнению системой управления базами данных операций обновления базы данных, поскольку сокращается число проверок и вспомогательных действий, поддерживающих целостность базы данных
- При проектировании реляционной базы данных почти всегда добиваются второй нормальной формы всех входящих в базу данных отношений
- В часто обновляемых базах данных обычно стараются обеспечить третью нормальную форму отношений
- На нормальную форму Бойса-Кодда внимание обращают гораздо реже, поскольку на практике ситуации, в которых у отношения имеется несколько составных перекрывающихся возможных ключей, встречаются нечасто