

Нормализация

(первые нормальные формы
в реляционной модели)

Бессарабов Н.В.

bes@fpm.kubsu.ru

2015 г.

Цели лекции

Теперь, когда мы уже знакомы с реляционной алгеброй и понимаем предназначение теоремы Хиса, можно приступить к изучению процессов нормализации схемы базы данных.

Что вспомнить: Понятия сильной и слабой сущностей (тема 2).

Мы сформируем понятие аномалии – несоответствия между ограничениями целостности концептуальной и логической (а также физической) схем данных. Цель нормализации это как раз устранение аномалий по включению, обновлению и удалению данных.

Будут рассмотрены четыре первые нормальные формы (1-я, 2-я, 3-я и НФБК). В одну группу они объединяются потому, что их определения основаны на классическом понятии функции, заданной на схеме отношения и на теореме Хиса.

Ещё две нормальные формы (четвёртая и пятая) потребуют модифицировать определение функциональной зависимости. Последняя нормальная форма домен-ключ знаменует возвращение к истокам – логическому подходу к реляционной теории.

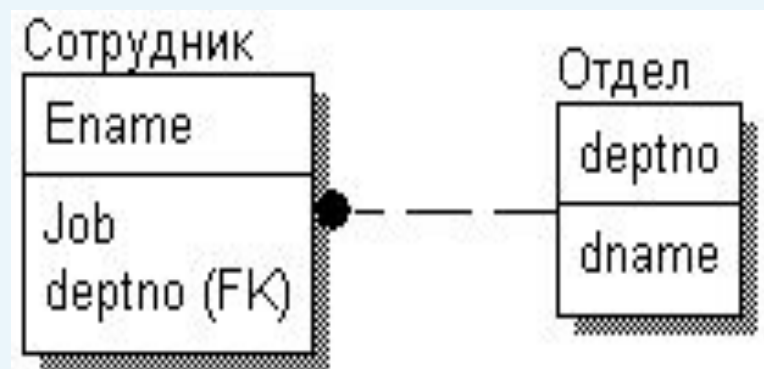
В заключение будет рекомендован способ получения схемы базы в первых четырех нормальных формах, почти всегда пригодный и достаточный для создания “правильных” баз данных.

Связи и внешние ключи

Реляционная модель бедна типами связей. Это один из главных ее недостатков. Используются связи видов “один-к-одному” (1:1) и “один-ко-многим” (1:N). Вы уже знаете, что связи могут иметь другие свойства концов кроме их мощности, например, обязательность, но в реляционной модели выразить их нельзя (нет таких слов). Поэтому первые нормальные формы будем рассматривать в рамках модели “сущность-связь”.

В этом разделе и в реляционной модели и в ER- диаграммах будем рассматривать только те связи между отношениями/сущностями которые в РМД образуются ссылочным ограничением целостности, называемым “внешний ключ” (“Foreign Key” – сокращённо FK).

Пример: Каждый сотрудник обязательно причисляется к одному из отделов. В отношении “Сотрудник” нельзя задать номер отдела deptno, не существующий в списке отделов (сущность “Отдел”). В одном отделе может быть ни одного, один, два и более сотрудников. Итак, имеем связь 1:N (“ко-многим” на стороне отношения “Сотрудник”).



Идентифицирующие и неидентифицирующие связи

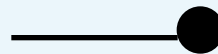
Типы связи **идентифицирующая** и **неидентифицирующая** относятся не к теории реляционных баз данных, а к стандарту моделирования IDEF1X, на котором основан ERWin (AllFusion Data Modeller). Этот инструмент мы используем в практических занятиях.

Если создается зависящая, то есть слабая, сущность, то внешний ключ передается в группу атрибутов, образующих первичный ключ. В этом случае используется идентифицирующая связь.

Неидентифицирующая связь соединяет две сущности передавая ключ в область неключевых атрибутов.

Обозначения:

Связь идентифицирующая



Связь неидентифицирующая



Заметим, что упомянутый ранее термин “обязательность **конца связи**” это уже не из IDEF1X.

Обязательность связей

Для неидентифицирующей связи можно указать обязательность (всей связи, а не её конца). Если связь обязательна (в ERWin признак No Nulls), то атрибуты внешнего ключа получают признак NOT NULL, означающий недопустимость неопределённых значений. Для необязательной связи (признак Nulls Allowed) внешний ключ может принимать значение NULL.

Обозначения:

1. Обязательная неидентифицирующая связь 

2. Необязательная неидентифицирующая связь помечается прозрачным ромбом со стороны родительской сущности



Зачем усложнять модель данных

Добавление понятий **сильной** и **слабой сущностей**, **идентифицирующей** и **неидентифицирующей связей**, **обязательности** и **необязательности** в **неидентифицирующих связях** существенно усложняет семантику используемой модели данных и добавляет проблем студентам начинающим осваивать базы данных.

Однако, уже при изучении нормализации вы увидите, что именно эти добавки позволяют получить адекватную модель данных для представления нормальных форм.

В дальнейшем мы сможем легко получать схему реляционной базы почти в законченном виде (позже мы эту мысль выразим точнее: “в третьей нормальной форме или нормальной форме Бойса-Кодда”). Для этого не нужно будет применять никаких формальных трудно осмысливаемых правил.

Замечание 1: Введённое расширение ER-модели это один из вариантов так называемой расширенной ER-модели (Extended ERD).

Замечание 2: Расширенная ER-модель (EERD) в ERWin реализует некий аналог наследования, так называемые категории.

Отличия реляционной модели и модели сущность - связь

Взаимно
однозначно
е
соответстви
е

Сущность		Отношение
Схема сущности	↔	Схема отношения
Атрибут сущности	↔	Атрибут отношения
Экземпляр сущности	↔	Кортеж отношения
Арность сущности	↔	Арность отношения
Сильная, слабая сущность		?
Связь арности больше 2		?
Связь идентифицирующая		?
Связь неидентифицирующая		?
Обязательность и необязательность неидентифицир. связи		?

Нет соответствующего
элемента

Пример первой нормальной формы (1НФ) – выравнивание сущностей

Ненормализованная сущность

Подписчик	Издание
Иванов	Правда, Известия, Коммерсант
Петров	ДАН

претендует на роль ключа

1НФ

Подписчик	Издание
Иванов	Правда
Иванов	Известия
Иванов	Коммерсант
Петров	ДАН

ключ

Неатомарный
столбец

На самом деле здесь
непервая нормальная
форма

Приведенный способ получения 1НФ называется “выравнивание сущностей”.
Переходя в реляционную модель мы будем также говорить о выравнивании отношений.

Определения 1НФ

Определение 1НФа (через атрибуты): Отношение/сущность находится в 1НФ, если значения всех его атрибутов атомарны.

Определение 1НФк (через ключи): Отношение/сущность находится в 1НФ, если оно имеет ключ.

Утверждение: 1НФа \Rightarrow 1НФк.

В самом деле, по определению реляционного отношения кортежи в отношении не повторяются. Если еще атрибуты атомарны, то есть удовлетворяют требованию предъявляемому в реляционной теории, то ключ в крайнем случае образуют все атрибуты, то есть ключ всегда существует.

Замечание 1: Обратное утверждение не верно.

Замечание 2: В дальнейшем для изложения и уточнения фактов реляционной модели используем язык EER-модели. Иначе говоря, мы позволим себе употреблять термины “сущность” и “отношение” как синонимы и, главное, рассуждать об отношениях используя термины “идентифицирующая связь”, “неидентифицирующая связь” и т.д.

Правила приведения к 1НФ

(способ выделения в отдельное отношение)

- Разделить составные атрибуты (в примере это “Дата зачисления и увольнения”) на простые (атомарные) (“Дата зачисления” и “Дата увольнения ”)
- Выделить “повторяющиеся” (однотипные, близкие) атрибуты (в примере это “Хобби” и “Тлф”)
- Для каждой такой группы атрибутов создать новую справочную сущность/отношение с одним атрибутом для повторяющейся группы
- Перенести в нее все значения повторяющихся атрибутов
- Установить **идентифицирующую связь** типа 1:N от исходной сущности к каждой созданной справочной сущности

Почему связь должна быть идентифицирующей?

Потому, что выделенные справочные сущности только уточняют свойства основной сущности и без привязки к экземпляру основной сущности их смысл теряется.

Например, на следующем слайде справочная сущность “хобби” имеет смысл “хобби данного сотрудника”, а без привязки её смысл “хобби вообще”, что не соответствует смыслу исходного отношения.

Пример приведения к 1НФ

(способ выделения в отдельное отношение/сущность)

Ненормализованное отношение

Отношения в 1НФ

Сотрудник

Табельн. номер
Фамилия
Имя
Отчеств
Должность
Хобби_1
Хобби_2
Оклад
Тлф_1
Тлф_2
Тлф_3
Дата зачисления или увольнения

Сотрудник

Табельн. номер
Фамили
Имя
Отчеств
Должность
Окла
Дата зачисления
Дата увольнения

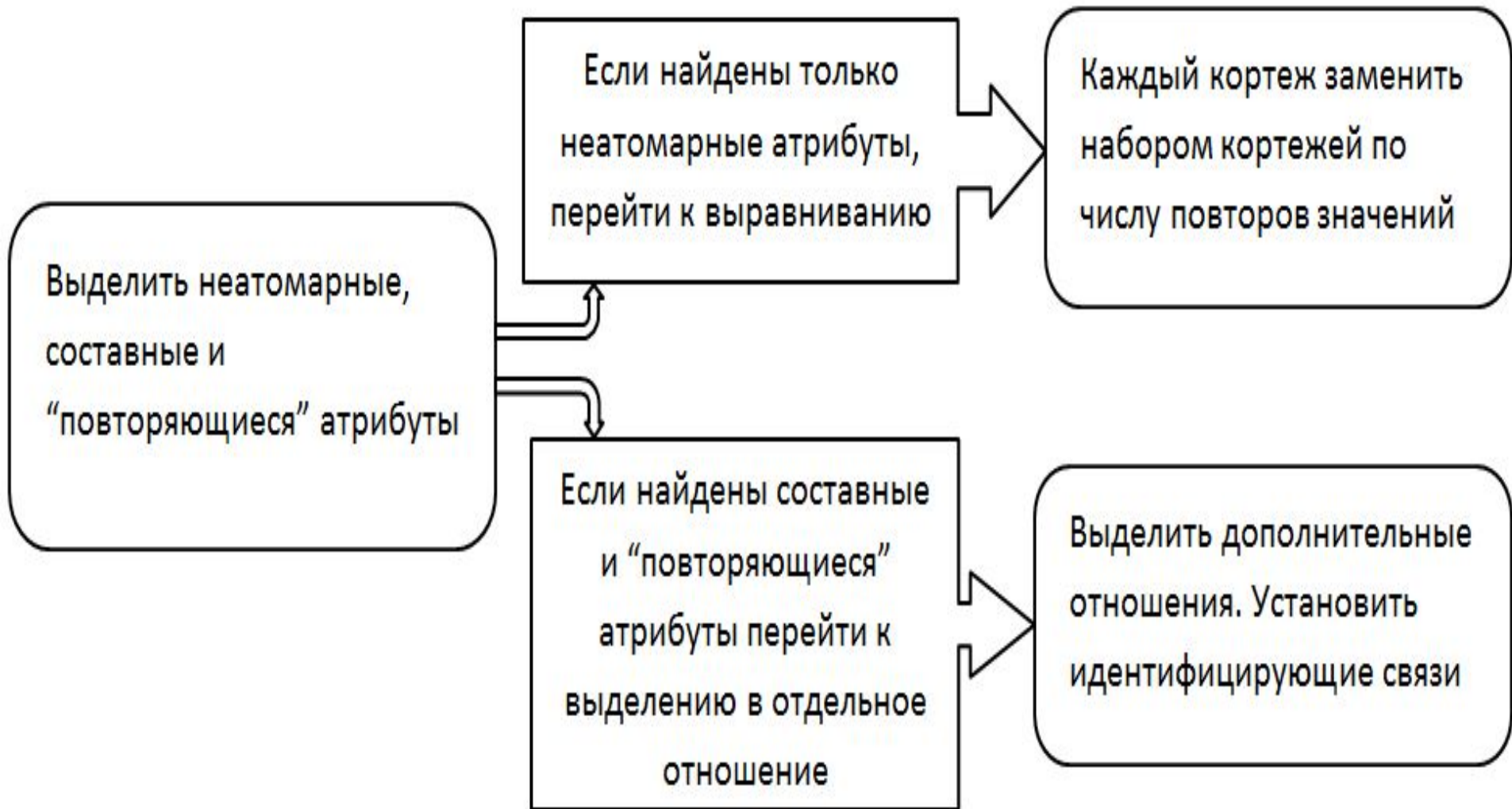
Хобби

Табельн. номер (FK)
Хобби

Телефон

Табельн. номер (FK)
Телефон

Два способа приведения к 1НФ



Сильные и слабые сущности

Вспомним, что сущности (и отношения) бывают двух видов: **слабые (зависимые) и сильные (независимые)**.

Сильная сущность существует “сама по себе”.

Первичные ключи у нее тоже свои, то есть определяются только своими полями (свойствами). На ER-диаграммах в ERWin сильные сущности представляются прямоугольниками.

Слабая сущность для идентификации своих экземпляров требует привязки к экземпляру связанной с ней, другой сильной или слабой, сущности. Первичный ключ слабой сущности использует поля этой связанной с ней сущности. Происходит миграция первичного ключа сильной сущности в первичный ключ связанной слабой сущности. На ER-диаграмме в ERWin слабая сущность представляется прямоугольником с закругленными углами.

На предшествующем слайде сущности “хобби” и “телефон” слабые, а преобразованная сущность “сотрудник” сильная.

Замечание: Не следует считать, что слабая сущность связывается только с сильной сущностью.

Замечание о непервой нормальной форме (Н1НФ, NFNF, NF²)

- 1НФ удовлетворяющая условию 1НФк, но не удовлетворяющая условию 1НФа называется **непервой нормальной формой (Н1НФ, NFNF, NF²)**.
- Основное преимущество модели, развиваемой на основе Н1НФ, в том, что хранить в базе можно значения не только атомарных, но и конструируемых, в том числе, реляционно-значных типов. В частности, отношения могут содержать вложенные отношения. Тем самым устраняется один из основных недостатков реляционного подхода – отсутствие агрегатов.
- В этом курсе мы вспомним Н1НФ при изучении объектных моделей.

Аномалии

Вспомним, что в рамках концептуальной модели определяется набор ограничений целостности. Часть из них может быть выражена в логической модели, а часть нет. Отсутствие в логической модели образов ограничений концептуальной модели определяет аномалии, проявляющиеся, в частности, при удалении, обновлении и введении данных.

Аномалия может пониматься как несоответствие бизнес-правил правилам работы в модели, логической или физической.

Можно понимать аномалии как несоответствия семантики двух связанных моделей данных.

Невозможность отображения ограничений может быть связана с особенностями разработанной схемы базы, а может определяться выбранной моделью данных. Например, в реляционной модели не реализуются ограничения, требующие разбора значений атрибутов, которые считаются атомарными.

Замечание: Приведенное здесь представление об аномалиях недостаточно для работы с современными реализациями баз данных в составе информационных систем, когда необходимо, например, учесть сетевой характер базы, особенности клиентской части и т. Д.₄₅

Пример проектирования через контекст

Рассмотрим в качестве примера странное медицинское учреждение. В нём есть врачи, к нему обращаются пациенты, которым назначаются лекарства и физиотерапевтические процедуры. Фиксируются побочные эффекты от применения назначенных лекарств.

Странностей у нашего учреждения много. В нём не интересуются диагнозом. Из временных (темпоральных) атрибутов фиксируется только время выполнения процедуры. Неплохо было бы кроме даты процедуры отмечать дату приёма врача на работу и не только это.

При подходе сверху вниз все атрибуты собираем в единственное отношение, которое называется контекстом в теории называемой формальным анализом концептов (Formal Concept Analysis).

Замечание 1: В реальных приложениях число атрибутов в контексте многие сотни и тысячи. Анализировать такую конструкцию практически не возможно. Поэтому выбран примитивный пример, что не мешает нам разобраться с аномалиями.

Замечание 2: К контекстам мы ещё вернёмся рассматривая Ортогональность сущностей.

Аномалии включения, удаления и модификации.

- Исходное отношение

Номер пациента	Номер врача	Дата	ФИО пациента	Адрес пациента	ФИО врача	Упражнение	Лекарств. препарат	Побочн. эффекты
----------------	-------------	------	--------------	----------------	-----------	------------	--------------------	-----------------

- Декомпозированное отношение

Врач

Номер_врача	ФИО_врача
-------------	-----------

Пациент

Номер_пациента	ФИО_пациента	Адрес_пациента
----------------	--------------	----------------

Врач_и_пациент

Номер_пациента	Номер_врача	Дата	Упражнение	Лекарств_препарат	Побочн_эфф
----------------	-------------	------	------------	-------------------	------------

Это отношение и есть контекст

Это и далее просто отношения

Аномалии включения, удаления и обновления для исходного отношения

а) Аномалии включения:

- если с больным ничего не делалось, то сведения о нем можно ввести только в виде фиктивных записей;
- если врач ничего не делал, то сведения о нем можно вести только в виде фиктивных записей;

б) Аномалии удаления:

- если сведения о закончившем курс пациенте удаляются, то могут быть удалены сведения о единственном враче с ним работавшем;
- если сменить препарат, то удаляются сведения о побочном эффекте, которые могут быть уникальными

в) Аномалии обновления:

- если курс лечения периодически повторяется и если между сеансами пациент меняет адрес, то первый найденный адрес может быть устаревшим.

Аномалии преобразованной схемы

В преобразованном наборе отношений “Врач”, “Пациент” и “Врач_и_пациент” часть аномалий устранена:

- можно ввести сведения о пациенте, который не проходил лечение и сведения о враче, который ничего не делал;
- после удалений сведений о больном остаются сведения о враче;

Изменение адреса больного затронет только отношение “Пациент”, но сама аномалия останется. Устраните её сами.

Часть аномалий устранена потому, что были выделены две сущности: “Врач”, “Пациент”.

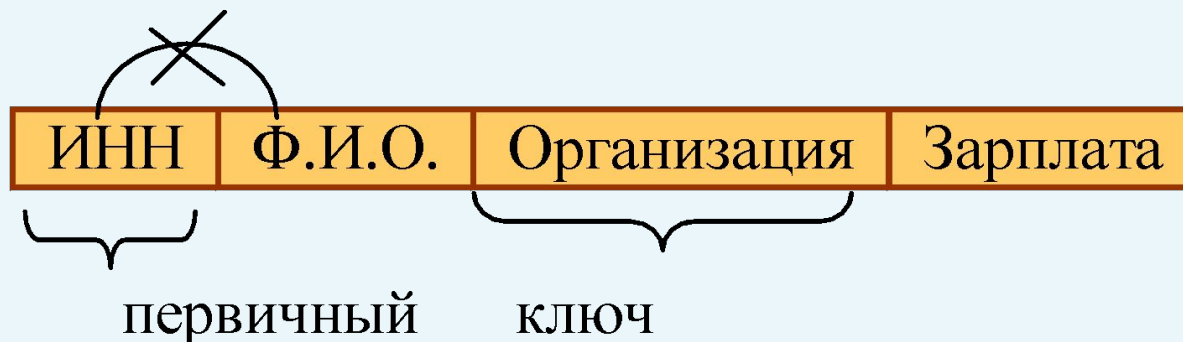
Снова о времени: Мы избегали введения необходимых темпоральных атрибутов только для того, чтобы упростить изложение. Но их отсутствие делает нашу схему совершенно не реалистичной, по-хорошему следовало бы отмечать времена выполнения всех изменений.

Запомните: **Цель нормализации – устранение аномалий** (а не уменьшение избыточности данных, как иногда говорят). Приведение к 1НФ может увеличить избыточность.

Вторая нормальная форма. Зависимости от части ключа

Может оказаться, что кроме функциональной зависимости всех ключевых атрибутов от **всего_ключа**, существуют зависимости неключевых атрибутов от **части ключа**.

Пример: В приведенном ниже отношении “Доходы_совместителей” поле “ИНН” однозначно определяет поле “Ф.И.О.”:



Определения 2НФ

Определение: Если набор атрибутов $\mathbf{B} = \{B_j\}$ зависит от всего набора атрибутов $\mathbf{A} = \{A_i\}$, но не зависит от части этого набора, то говорят, что функциональная зависимость $\mathbf{f}: \mathbf{A} \rightarrow \mathbf{B}$ полная (или, в другой терминологии, неприводимая слева).

Определение 2НФа (через атрибуты): Отношение в 1НФ находится в 2НФ, если ни один атрибут вне первичного ключа не находится в функциональной зависимости от атрибутов, образующих часть ключа.

Определение 2НФк (через ключи): Отношение в 1НФ находится в 2НФ, если каждый неключевой атрибут, находится в полной функциональной зависимости от ключа.

Замечание 1: Определения 2НФа и 2НФк эквивалентны.

Замечание 2: Если единственный ключ отношения в 1НФ является простым (не конкатенированным), то отношение находится в 2НФ.

Правило приведения к 2НФ



- Выделить неключевые атрибуты, зависящие от части первичного ключа. Иначе говоря, найти функциональную зависимость группы неключевых атрибутов от части атрибутов ключа.
- Создать новую сущность. В соответствии с теоремой Хиса все ее атрибуты входят в найденную выше функциональную зависимость.
- Вычеркнуть атрибуты-значения найденной функции в исходной сущности. Почему идентифицирующую?
- Установить **идентифицирующую связь 1:N** или **N:1** от исходной сущности к созданной сущности.

В примере на следующем слайде существует зависимость атрибутов “Фамилия”, “Имя”, “Отчество”, “Должность” от атрибута “Таб_номер_рук”, являющегося частью первичного ключа.

Пример приведения к 2НФ (в ERWin)



Ненормализованное
отношение

Проек


 Наименование проекта
 Таб_номер_рук
Дата Начала Завершения
Имя Отчеств Должность

Отношение в 2НФ

Проект

 Наименование проекта
 Рук., Табельный номер (FK)
Дата начала Дата завершения

Руководитель


 Табельный номер
Фамилия Имя Отчество Должность

Третья нормальная форма. Зависимости неключевых атрибутов

Кроме (1) функциональной зависимости всех атрибутов от всего ключа и (2) зависимостей неключевых атрибутов от части ключа могут существовать (3) зависимости неключевых атрибутов от других неключевых атрибутов.

Пример: Функция f : Должность \rightarrow Оклад (если оклад зависит только от должности)

Сотрудн

 Табельный номер
Фамили
Имя
Отчеств
Должност
Окла
Д

Определения 3НФ

Определение (транзитивной и прямой ФЗ): функциональная зависимость $A \rightarrow C$ называется транзитивной, если найдется атрибут B такой, что $A \rightarrow B$, $B \rightarrow C$ функциональные зависимости. Если не существует транзитивной зависимости, то функциональная зависимость называется прямой.

Почему не во второй?

Определение 3НФк (через ключи): отношение в 1НФ находится в 3НФ, если все его атрибуты прямо зависят от ключа.

Почему не во второй?

Определение 3НФа (через атрибуты): отношение в 1НФ находится в 3НФ, если оно не содержит зависимостей неключевых атрибутов от других атрибутов, не входящих в состав первичного ключа.

Замечание: Определения 3НФк и 3НФа эквивалентны.

Правило приведения к 3НФ

- Найти функциональную зависимость неключевых атрибутов от других неключевых атрибутов.
- Создать новую сущность. В соответствии с теоремой Хиса все ее атрибуты входят в найденную функциональную зависимость.
- Вычеркнуть атрибуты-значения найденной функции в исходной сущности.
- Установить **неидентифицирующую** связь от созданной сущности к исходной сущности

Почему связь неидентифицирующая?

Потому, что аргумент выделенной функции не содержит ключевых столбцов исходного отношения. Это означает что создаваемое справочное отношение содержит сведения общие для всех экземпляров исходного отношения.

Пример приведения к 3НФ (в ERWin)

Пример: Функция f: Должность → Оклад

Ненормализованное
отношение

Сотрудн

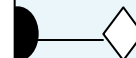


Нормализованное
отношение

Сотрудн



Должно



Теорема. Если отношение находится в 3НФ, то оно находится во 2НФ.

Предварительно сформулируем два отрицательных высказывания.

Нарушение условия 2НФ. Во 2НФ каждый непервичный атрибут не может зависеть от части ключа. Обозначим нарушение условия $\neg(2НФ)$.

Нарушение условия 3НФ. В 3НФ ни один из непервичных атрибутов не может транзитивно зависеть от ключа. Обозначим нарушение $\neg(3НФ)$.

Выбор схемы доказательства: Достаточно показать, что из частичной зависимости следует транзитивная зависимость. Это будет означать, что из нарушения условия 2НФ следует нарушение условия 3НФ. В самом деле, по определению импликации $x \Rightarrow y \equiv (\neg x) \vee y$. Докажем, что $((3НФ) \Rightarrow (2НФ)) \equiv (\neg(2НФ) \Rightarrow \neg(3НФ))$. В самом деле, в более простых обозначениях x и y имеем

$$(\neg x \Rightarrow \neg y) \equiv (\neg(\neg x) \vee (\neg y)) \equiv x \vee (\neg y) \equiv (\neg y) \vee x \equiv y \Rightarrow x.$$

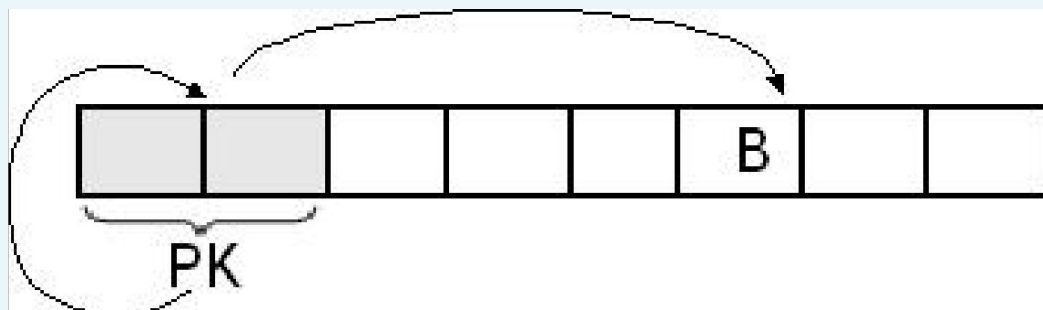
Доказательство: Пусть $\neg(2НФ)$, то есть непервичный атрибут A частично зависит от ключа K . Это означает, что $\exists K' \subset K: K' \rightarrow A$.

Конечно, не существует зависимости $K' \rightarrow K$, иначе K' было бы ключом. Ведь ключ по определению минимален.

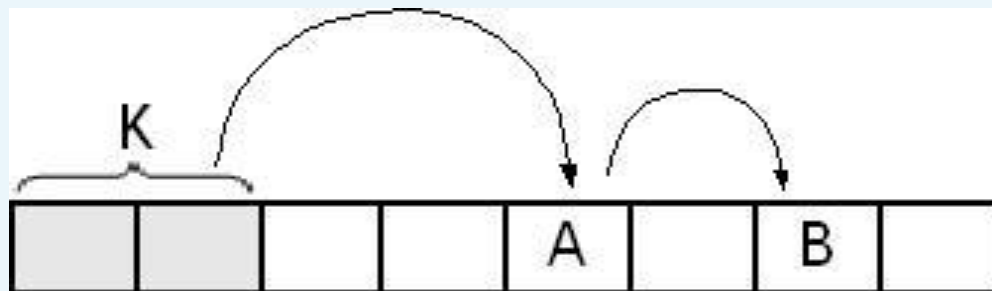
Итак, существуют $f1: K \rightarrow K'$, $f2: K' \rightarrow A$, то есть цепочка $f1, f2$ транзитивная, ч.т.д.

Графическое пояснение к теореме $3НФ \Rightarrow 2НФ$

Цепочка транзитивных ФЗ для 2НФ



Цепочка транзитивных ФЗ для 3НФ



Теорема Хиса. Контрпример.

Пример: В отношении имеются атрибуты “зарплата” и “допустимые комиссионные”, которые вычисляются по некоторой формуле исходя из зарплаты. Выделив функцию “зарплата” \implies “допустимые комиссионные” получаем некоторый набор экземпляров сущности, но при вводе следующего экземпляра необходимо будет снова обратиться к определению функции..

Вывод: В правилах нормализации необходимо помнить, что образованные сущности должны иметь приемлемую семантику.

Отношения в 3НФ, имеющие несколько ключей

Пример 1: Отношение с тремя не пересекающимися ключами.
Атрибут Табельный_Номер уникальный.

PK1		PK2		PK3	
Табельный_Номер	ИНН	ФИО	Дата_рождения	Паспорт. данные

Пример 2: Отношение с двумя пересекающимися ключами

Бригада	Стажер	Наставник
1	Иванов	Живаго
1	Петров	Петраго
2	Сидоров	Соловаго
2	Иванов	Соловаго

PK1 PK1 PK2 PK2

Нормальная форма Бойса-Кодда (Boyce-Codd)

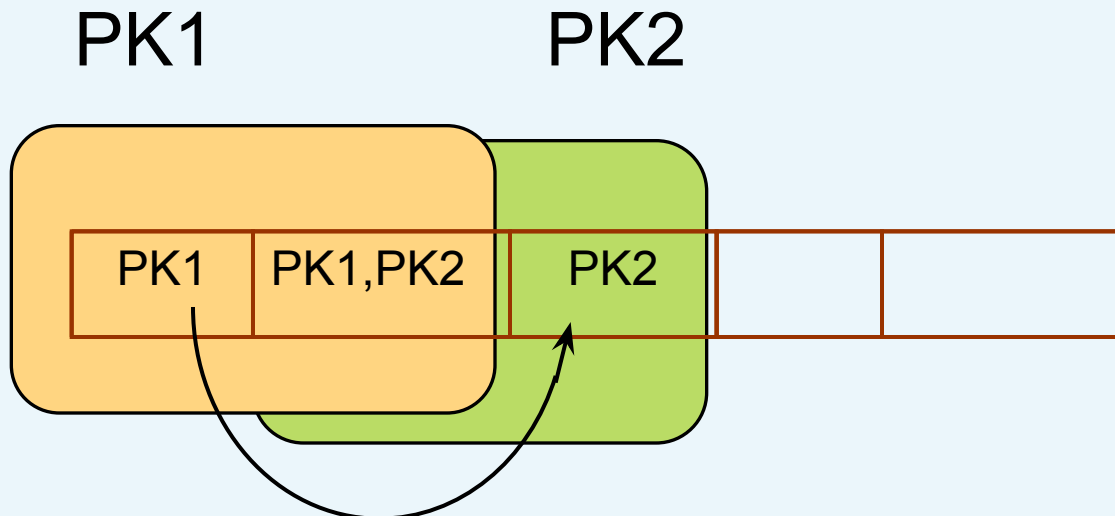
Исходное определение 3НФ основывается на предположении о том, что первичный ключ единственный. Может оказаться, что:

- отношение имеет два или более ключа-кандидата (альтернативных ключа);
- по крайней мере два из них конкатенированы;
- некоторые из конкатенированных ключей перекрываются (имеют общие атрибуты).

В этом случае после получения 3НФ необходимо привести отношения к нормальной форме Бойса-Кодда, сокращённо, НФБК. Ещё её называли исправленной третьей нормальной формой.

Какие функциональные зависимости исследуют при приведении к НФБК?

Ответ: Если ранее приводили отношение к 1НФ и 2НФ, то только функции, действующие из одного ключа в не принадлежащие этому первому ключу атрибуты второго пересекающегося с ним ключа



Виды функциональных зависимостей

Определение 1 (Тривиальная функциональная зависимость): ФЗ $f: A \rightarrow B$ тривиальна тогда и только тогда, когда правая часть функциональной зависимости является подмножеством (не обязательно собственным) левой части, то есть $B \subseteq A$.

Повторим также:

Определение 2 (Функциональная зависимость неприводимая слева): ФЗ называется неприводимой слева, если ни один атрибут в левой части не может быть опущен без разрушения функциональной зависимости.

Определения НФБК

Определение 1 (НФБК): Отношение находится в НФБК тогда и только тогда, когда каждая нетривиальная и неприводимая слева функциональная зависимость имеет аргументом ключ.

Определение 2 (НФБК): Отношение находится в НФБК тогда и только тогда, когда аргументы любой функциональной зависимости есть ключи.

Чего не должно быть в отношении находящемся в НФБК:
Если ранее приводили отношение к 1НФ и 2НФ, то функциональных зависимостей, действующих из атрибутов принадлежащих только одному из пересекающихся ключей в атрибуты принадлежащие только другому ключу.

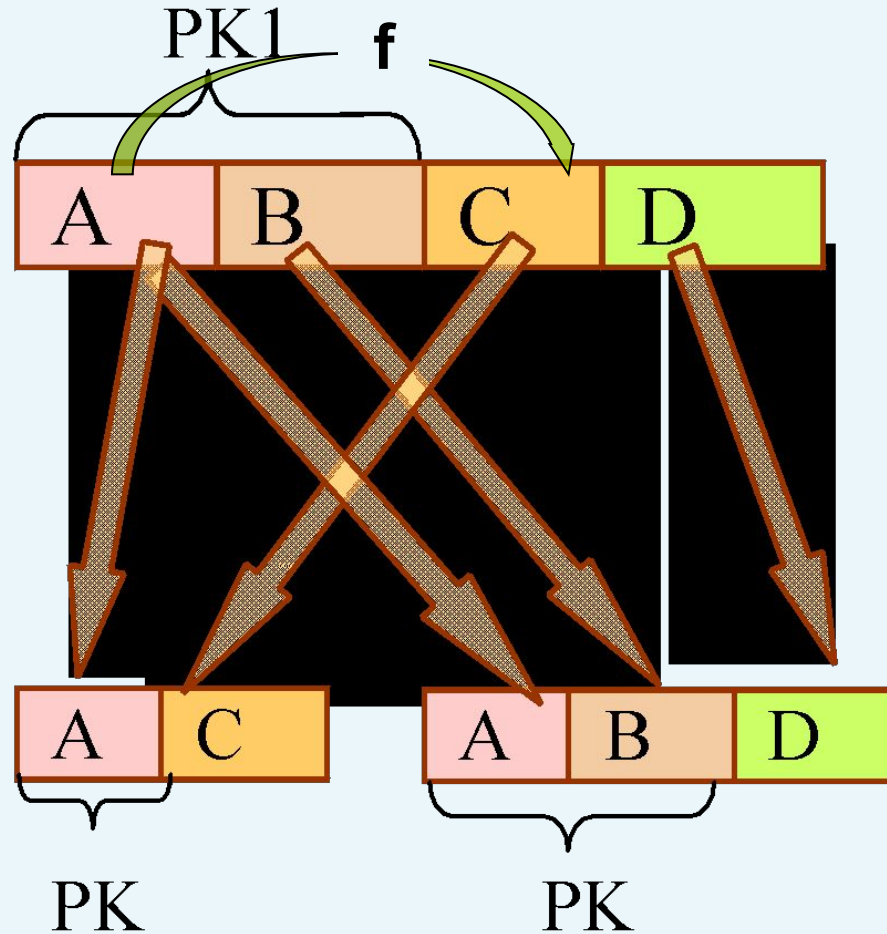
Правила преобразования в НФБК

Совпадают с правилами для ЗНФ.

Связь между образовавшимися сущностями также неидентифицирующая.

Отличия только в анализируемых функциях.

Мнемоника преобразования для НФБК (зависимость $f:A \rightarrow C$)



Пример преобразования в НФБК

- Заменяем отношение двумя проекциями: "Бригада - Стажёр" и "Стажёр - Наставник".

Бригада	Наставник
1	Живаго
1	Петраго
2	Соловаго

РК

Стажёр	Наставник
Иванов	Живаго
Петров	Петраго
Сидоров	Соловаго
Иванов 2-й	Соловаго

РК

FK

Обратите внимание на
переименование Иванова из 2-й
бригады

Теорема: Любое отношение с двумя атрибутами находится в НФБК

Пусть атрибуты поименованы как A и B . Возможны четыре случая:

- $\{A, B\}$ единственный ключ, то есть нетривиальных зависимостей нет.
- Имеется единственная нетривиальная зависимость $A \rightarrow B$, но нет $B \rightarrow A$. Единственный ключ A и единственная функция $A \rightarrow B$ содержит A слева. (*)
- $B \rightarrow A$, но нет $A \rightarrow B$. Симметричная (*) ситуация.
- И $A \rightarrow B$ и $B \rightarrow A$. Значит A и B не пересекающиеся ключи. Других функций нет.

Замечание: Вывод о том, что следует все отношения декомпозировать до двухатрибутных – неверен.

Причина: декомпозиция может оказаться неполной.

Нормальная форма схемы базы

Определение: Говорят, что схема базы данных находится в нормальной форме nНФ, если каждое ее отношение находится в этой нормальной форме или в форме которая «не слабее» nНФ.

Сходимость процесса нормализации по теореме Хиса

Процессы нормализации до первых четырёх нормальных форм (1НФ, 2НФ, 3НФ, НФБК) сходятся.

В самом деле, приведение к 1НФ либо не меняет число столбцов (выравнивание) либо однократно увеличивает его на количество составных атрибутов.

При переходе к следующим формам (2НФ, 3НФ, НФБК) каждая декомпозиция по теореме Хиса приводит к отношениям с числом атрибутов меньшим по крайней мере на единицу, а число отношений схемы и исходное число атрибутов в каждом отношении схемы по определению конечно. Поэтому число создаваемых отношений также конечно.

О стиле проектирования базы

В этой лекции неявно был принят один из возможных стилей проектирования базы данных при котором все атрибуты, используемые в схеме базы, относятся к единственному отношению, называемому иногда универсальным. А уже в нем ищут функциональные зависимости и, используя теорему Хиса, выделяют отношения, которые не могут быть декомпозированы.

Более эффективным может быть проектирование, основанное на построении ER-диаграммы с выделением не декомпозлируемых сильных и слабых сущностей. При достаточном опыте можно сразу получить схему в третьей нормальной форме. Невозможность дальнейшей декомпозиции определяется по отсутствию функциональных зависимостей кроме зависимостей от первичных ключей.

Нормализация в ER-модели

Отношению в реляционной модели данных соответствует сущность в ER-диаграмме. Любая реляционная связь представима связью в ER-модели.

Из наличия такого отображения следует возможность переноса понятия и алгоритма нормализации из реляционной модели в модель «сущность-связь».

Обратный переход от ER- диаграмм к реляционной модели связан с потерей некоторой части информации и, может быть замены одной связи двумя и более. Последнее определяется тем, что связи в реляционной модели более примитивны.

Простой способ получения отношений сразу в 3НФ и уточнения до НФБК:

1. Выделите простые сущности с атомарными атрибутами, не содержащие в себе другие сущности и не имеющие составных атрибутов и групп однородных атрибутов. Если этот этап выполнен правильно, получена 3НФ.
2. Уточните ключевые атрибуты, выделив все альтернативные ключи. Чтобы окончательно убедиться в простоте сущностей проверьте наличие ФЗ кроме зависимостей от ключей. Если они обнаружены, декомпозируйте такие сущности по Хису.
3. Если есть пересекающиеся ключи, проверьте условие: все ФЗ должны иметь аргументами первичные ключи. При обнаружении других зависимостей получите НФБК, используя теорему Хиса.

Заключение

Рассмотрены понятия связей между отношениями, сильные и слабые отношения (сущности), обязательность и необязательность связей, а также принятые в стандарте IDEF1X идентифицирующие и неидентифицирующие связи.

Рассмотрены аномалии по включению, удалению и обновлению данных.

Изученные четыре нормальных формы (1НФ, 2НФ, 3НФ и исправленная третья форма (НФБК)) составляют минимум, которым во многих случаях можно ограничиться в процессе нормализации схемы базы. Установлены соотношения между нормальными формами. Все рассмотренные алгоритмы нормализации основаны на теореме Хиса.

Процесс нормализации всегда сходится. Останавливать его следует, когда в отношениях останутся только функциональные зависимости от первичного ключа.

Основные понятия



Словарь студента 1/3

- ❑ Аномалия – несоответствие ограничений целостности модели бизнеса концептуального уровня и логической модели.
- ❑ Аномалии по включению, обновлению (модификации) и удалению – устраняются нормализацией. →
- ❑ Зависимость функциональная неприводимая слева – ни один из атрибутов аргумента нельзя удалить без разрушения зависимости.
- ❑ Зависимость функциональная прямая – $f:A \rightarrow C$ существует, но не найдётся набора аргументов B , такого, что существуют функциональные зависимости $f_1:A \rightarrow B$ и $f_2:B \rightarrow C$.
- ❑ Зависимость функциональная транзитивная – $f:A \rightarrow C$ транзитивна, если найдётся набор аргументов B , такой, что $f_1:A \rightarrow B$ и $f_2:B \rightarrow C$ функциональные зависимости.
- ❑ Зависимость функциональная тривиальная – $f:A \rightarrow B$ тривиальна, если $B \subset A$, может быть несобственным образом.

Словарь студента 2/3

- ❑ Нормальная форма Бойса-Кодда – она же исправленная третья нормальная форма. Необходимость проверки возникает при наличии пересекающихся первичных ключей. Характеризуется отсутствием функциональных зависимостей, действующих из атрибутов, принадлежащих только одному из пересекающихся ключей, в атрибуты, принадлежащие только второму из пересекающихся ключей.
- ❑ Нормальная форма вторая – предполагается, что отношение уже находится в 1НФ или Н1НФ. Определяющее свойство – отсутствие зависимостей неключевых атрибутов от части ключа.
- ❑ Нормальная форма непервая (Н1НФ, NFNF, NF²) – имеет ключ, но атрибуты могут быть неатомарными.
- ❑ Нормальная форма первая (1НФ, 1NFNF) – имеет ключ и атомарные атрибуты.

Словарь студента 3/3

- ❑ Связь идентифицирующая. (Термин из стандарта IDEF1x). Устанавливается между независимой (родительской) и зависимой (дочерней) сущностями. Атрибуты первичного ключа родительской сущности мигрируют в состав первичного ключа дочерней сущности.
- ❑ Связь неидентифицирующая. (Термин из стандарта IDEF1x). Связывает независимые сущности. Атрибуты первичного ключа родительской сущности мигрируют в состав неключевых атрибутов дочерней сущности.
- ❑ Связь обязательная. Экземпляры хранимых в базе связываемых сущностей обязательно входят в связь.
- ❑ Сходимость процесса нормализации выполняемого за счет применения теоремы Хиса – процесс всегда сходится.
- ❑ Уровень нормализации схемы базы определяется низшим уровнем нормализации входящих в неё отношений.