

Теория нормальных форм

Курс «Базы данных»

Вадим Цесько

Санкт-Петербургский государственный политехнический университет

1 марта 2012 г.

Содержание

- 1 Функциональные зависимости
- 2 Нормализация
- 3 Первая нормальная форма
- 4 Вторая нормальная форма
- 5 Третья нормальная форма
- 6 Возможные ключи
- 7 Нормальная форма Бойса-Кодда
- 8 Четвёртая нормальная форма

Функциональная зависимость

Если даны два атрибута X и Y некоторого отношения, то говорят, что Y функционально зависит от X , если в любой момент времени каждому значению X соответствует ровно одно значение Y .

Обозначение: $X \rightarrow Y$ (Y функционально зависит от X).

Функциональная зависимость

Если даны два атрибута X и Y некоторого отношения, то говорят, что Y функционально зависит от X , если в любой момент времени каждому значению X соответствует ровно одно значение Y .

Обозначение: $X \rightarrow Y$ (Y функционально зависит от X).

Пример

Атрибуты функционально зависят от первичного ключа.

Избыточная функциональная зависимость

Может быть получена на основе других функциональных зависимостей БД.

Избыточная функциональная зависимость

Может быть получена на основе других функциональных зависимостей БД.

Корректная схема БД

Отсутствуют избыточные функциональные зависимости.

Нормализация

Избыточная функциональная зависимость

Может быть получена на основе других функциональных зависимостей БД.

Корректная схема БД

Отсутствуют избыточные функциональные зависимости.

Нормализация

Обратимый пошаговый процесс замены данной совокупности отношений другой схемой с устранением избыточных функциональных зависимостей.

Избыточная функциональная зависимость

Может быть получена на основе других функциональных зависимостей БД.

Корректная схема БД

Отсутствуют избыточные функциональные зависимости.

Нормализация

Обратимый пошаговый процесс замены данной совокупности отношений другой схемой с устранением избыточных функциональных зависимостей.

Обычно нормализация представляет собой декомпозицию отношений.

Условие обратимости

Сохранение эквивалентности схем БД:

- Не должны появляться ранее отсутствовавшие сущности (кортежи)
- На отношениях новой схемы должно выполняться исходное множество функциональных зависимостей

Условие обратимости

Сохранение эквивалентности схем БД:

- Не должны появляться ранее отсутствовавшие сущности (кортежи)
- На отношениях новой схемы должно выполняться исходное множество функциональных зависимостей

Было

- STUDENTS: (ID, NAME, GROUP_NUMBER, HEAD_NAME, SPECIALITY, ...)

Условие обратимости

Сохранение эквивалентности схем БД:

- Не должны появляться ранее отсутствовавшие сущности (кортежи)
- На отношениях новой схемы должно выполняться исходное множество функциональных зависимостей

Было

- STUDENTS: (ID, NAME, GROUP_NUMBER, HEAD_NAME, SPECIALITY, ...)

Стало

- STUDENTS: (ID, NAME, GROUP_ID, ...)
- GROUPS: (ID, NUMBER, HEAD_ID, SPECIALITY_ID, ...)
- SPECIALITIES: (ID, NAME, ...)

Пример ещё раз

- STUDENTS: (ID, NAME, GROUP_NUMBER, HEAD_NAME, SPECIALITY, ...)

Пример ещё раз

- STUDENTS: (ID, NAME, GROUP_NUMBER, HEAD_NAME, SPECIALITY, ...)

Что плохо:

Пример ещё раз

- STUDENTS: (ID, NAME, GROUP_NUMBER, HEAD_NAME, SPECIALITY, ...)

Что плохо:

- Дублирование данных

Пример ещё раз

- STUDENTS: (ID, NAME, GROUP_NUMBER, HEAD_NAME, SPECIALITY, ...)

Что плохо:

- Дублирование данных
- Аномалии при выполнении операций

Простой атрибут

Атрибут, значения которого **атомарны** (неделимы).

Первая нормальная форма

Простой атрибут

Атрибут, значения которого **атомарны** (неделимы).

Сложный атрибут

Объединение нескольких простых атрибутов, которые могут быть определены на одном или разных доменах.

Первая нормальная форма

Простой атрибут

Атрибут, значения которого **атомарны** (неделимы).

Сложный атрибут

Объединение нескольких простых атрибутов, которые могут быть определены на одном или разных доменах.

Первая нормальная форма (1NF)

Отношение находится в 1NF, если **все его атрибуты простые** (т. е. их значения атомарны).

Было

- STUDENTS: (FIRST_NAME, LAST_NAME, TASKS, ...)
- Атрибут TASKS может содержать от 0 до N элементов

Пример нормализации

Было

- STUDENTS: (FIRST_NAME, LAST_NAME, TASKS, ...)
- Атрибут TASKS может содержать от 0 до N элементов

Стало

- STUDENTS: (ID, FIRST_NAME, LAST_NAME, ...)
- TASKS: (STUDENT_ID, NAME, ...)

Более сложный пример нормализации

Было

- COURSES: (NAME, BOOK, ...)

Более сложный пример нормализации

Было

- COURSES: (NAME, BOOK, ...)

Стало

- COURSES: (ID, NAME, ...)
- BOOKS: (COURSE_ID, NAME, ...)

Более сложный пример нормализации

Было

- COURSES: (NAME, BOOK, ...)

Стало

- COURSES: (ID, NAME, ...)
- BOOKS: (COURSE_ID, NAME, ...)

Ещё лучше

- COURSES: (ID, NAME, ...)
- BOOKS: (ID, NAME, AUTHOR, ...)
- COURSE_BOOK: (COURSE_ID, BOOK_ID)

Вторая нормальная форма

Составной РК

РК, состоящий из нескольких атрибутов.

Вторая нормальная форма

Составной РК

РК, состоящий из нескольких атрибутов.

Функционально полная зависимость

Атрибут **функционально полно** зависит от составного РК, если он зависит от всего РК в целом и не зависит от какого-либо отдельного атрибута РК.

Вторая нормальная форма

Составной РК

РК, состоящий из нескольких атрибутов.

Функционально полная зависимость

Атрибут **функционально полно** зависит от составного РК, если он зависит от всего РК в целом и не зависит от какого-либо отдельного атрибута РК.

Вторая нормальная форма (2NF)

Отношение находится во **2NF**, если оно находится в **1NF** и каждый неключевой атрибут функционально полно зависит от РК.

Вторая нормальная форма

Составной РК

РК, состоящий из нескольких атрибутов.

Функционально полная зависимость

Атрибут **функционально полно** зависит от составного РК, если он зависит от всего РК в целом и не зависит от какого-либо отдельного атрибута РК.

Вторая нормальная форма (2NF)

Отношение находится во **2NF**, если оно находится в **1NF** и каждый неключевой атрибут функционально полно зависит от РК.

Для простого РК

$1NF \equiv 2NF$.

Было

- BOOKS: (NAME, EDITION, PUBLISHER, ...)
- PK: (NAME, EDITION)
- ФЗ: NAME → PUBLISHER

Пример нормализации

Было

- BOOKS: (NAME, EDITION, PUBLISHER, ...)
- PK: (NAME, EDITION)
- ФЗ: NAME \rightarrow PUBLISHER

Стало

- BOOKS: (ID, NAME, PUBLISHER, ...)
- EDITIONS: (BOOK_ID, EDITION, ...)
- PK: (BOOK_ID, EDITION)

Ещё один пример нормализации

Было

- EMPLOYEES: (NAME, SKILL, ADDRESS, ...)
- PK: (NAME, SKILL)
- ФЗ: NAME → ADDRESS

Ещё один пример нормализации

Было

- EMPLOYEES: (NAME, SKILL, ADDRESS, ...)
- PK: (NAME, SKILL)
- ФЗ: NAME → ADDRESS

Стало

- EMPLOYEES: (NAME, ADDRESS, ...)
- SKILLS: (NAME, SKILL, ...)
- PK: (NAME, SKILL)

Транзитивная зависимость

Если для атрибутов X , Y и Z некоторого отношения выполняется $X \rightarrow Y$ и $Y \rightarrow Z$, а обратные ФЗ отсутствуют, то Z транзитивно зависит от X .

Транзитивные зависимости

Транзитивная зависимость

Если для атрибутов X , Y и Z некоторого отношения выполняется $X \rightarrow Y$ и $Y \rightarrow Z$, а **обратные ФЗ отсутствуют**, то Z **транзитивно зависит** от X .

Ещё раз

Обратные ФЗ $Y \rightarrow X$ и $Z \rightarrow Y$ **отсутствуют**.

Транзитивные зависимости

Транзитивная зависимость

Если для атрибутов X , Y и Z некоторого отношения выполняется $X \rightarrow Y$ и $Y \rightarrow Z$, а обратные ФЗ отсутствуют, то Z транзитивно зависит от X .

Ещё раз

Обратные ФЗ $Y \rightarrow X$ и $Z \rightarrow Y$ отсутствуют.

Пример

- STUDENTS: (NAME, A_MARK_COUNT, GOT_RED_DIPLOMA, ...)
- ФЗ: NAME \rightarrow A_MARK_COUNT,
A_MARK_COUNT \rightarrow GOT_RED_DIPLOMA

Третья нормальная форма

Третья нормальная форма (3NF)

Отношение находится в **3NF**, если оно находится во **2NF** и все неключевые атрибуты **нетранзитивно зависят** от первичного ключа.

Третья нормальная форма

Третья нормальная форма (3NF)

Отношение находится в **3NF**, если оно находится во **2NF** и все неключевые атрибуты **нетранзитивно зависят** от первичного ключа.

Сумма требований к отношению в 3NF:

Третья нормальная форма (3NF)

Отношение находится в **3NF**, если оно находится во **2NF** и все неключевые атрибуты **нетранзитивно зависят** от первичного ключа.

Сумма требований к отношению в 3NF:

- Атомарность атрибутов (из 1NF)

Третья нормальная форма (3NF)

Отношение находится в **3NF**, если оно находится во **2NF** и все неключевые атрибуты **нетранзитивно зависят** от первичного ключа.

Сумма требований к отношению в 3NF:

- Атомарность атрибутов (из 1NF)
- Зависимость неключевых атрибутов от всего ключа в целом (из 2NF) — подразумевается зависимость неключевых атрибутов от ключа

Третья нормальная форма (3NF)

Отношение находится в **3NF**, если оно находится во **2NF** и все неключевые атрибуты **нетранзитивно зависят** от первичного ключа.

Сумма требований к отношению в 3NF:

- Атомарность атрибутов (из 1NF)
- Зависимость неключевых атрибутов от всего ключа в целом (из 2NF) — подразумевается зависимость неключевых атрибутов от ключа
- Нетранзитивность этих зависимостей

Было

- STUDENTS: (NAME, A_MARK_COUNT, GOT_RED_DIPLOMA, ...)

Пример нормализации

Было

- STUDENTS: (NAME, A_MARK_COUNT, GOT_RED_DIPLOMA, ...)

Стало

- STUDENTS: (NAME, A_MARK_COUNT, ...)
- RED_DIPLOMA: (A_MARK_COUNT, RED_DIPLOMA, ...)

Ещё один пример нормализации

Было

- WINNERS: (TOURNAMENT, YEAR, PLAYER_NAME, BIRTHDATE, ...)
- ФЗ: (TOURNAMENT, YEAR) \rightarrow PLAYER_NAME,
PLAYER_NAME \rightarrow BIRTHDATE

Ещё один пример нормализации

Было

- WINNERS: (TOURNAMENT, YEAR, PLAYER_NAME, BIRTHDATE, ...)
- ФЗ: (TOURNAMENT, YEAR) \rightarrow PLAYER_NAME,
PLAYER_NAME \rightarrow BIRTHDATE

Стало

- PLAYERS: (ID, NAME, BIRTHDATE, ...)
- WINNERS: (TOURNAMENT, YEAR, PLAYER_ID, ...)

- РК должен **однозначно определять** каждый кортеж в отношении

- РК должен **однозначно определять** каждый кортеж в отношении
- Все атрибуты отношения должны быть **атомарны** до той степени, которая обеспечивается СУБД

- РК должен **однозначно определять** каждый кортеж в отношении
- Все атрибуты отношения должны быть **атомарны** до той степени, которая обеспечивается СУБД
- Каждый неключевой атрибут должен **непосредственно зависеть** от РК

- РК должен **однозначно определять** каждый кортеж в отношении
- Все атрибуты отношения должны быть **атомарны** до той степени, которая обеспечивается СУБД
- Каждый неключевой атрибут должен **непосредственно** зависеть от РК
- Каждый неключевой атрибут должен зависеть от **всего** РК

- РК должен **однозначно определять** каждый кортеж в отношении
- Все атрибуты отношения должны быть **атомарны** до той степени, которая обеспечивается СУБД
- Каждый неключевой атрибут должен **непосредственно** зависеть от РК
- Каждый неключевой атрибут должен зависеть от **всего** РК
- Каждый неключевой атрибут не должен зависеть **ни от чего другого** кроме РК

Возможный (альтернативный) ключ (АК)

Атрибут или несколько атрибутов, которые отвечают требованиям, предъявляемым к РК:

- Обеспечивает уникальность кортежей
- Является избыточным (не существует подмножества атрибутов РК, которые обеспечивают уникальность кортежей)

Возможный (альтернативный) ключ (АК)

Атрибут или несколько атрибутов, которые отвечают требованиям, предъявляемым к РК:

- Обеспечивает уникальность кортежей
- Является избыточным (не существует подмножества атрибутов РК, которые обеспечивают уникальность кортежей)

Отношение может иметь несколько возможных ключей (АК).

Возможный (альтернативный) ключ (АК)

Атрибут или несколько атрибутов, которые отвечают требованиям, предъявляемым к РК:

- Обеспечивает уникальность кортежей
- Является избыточным (не существует подмножества атрибутов РК, которые обеспечивают уникальность кортежей)

Отношение может иметь несколько возможных ключей (АК).

В качестве РК выбирается один из АК — обычно тот, который включает наименьшее число полей, либо в качестве РК используется суррогатный ключ.

Нормальная форма Бойса-Кодда (BCNF)

Отношение находится в **BCNF**, если оно находится в **3NF** и в нём отсутствуют зависимости атрибутов первичного ключа от неключевых атрибутов.

Нормальная форма Бойса-Кодда (BCNF)

Отношение находится в **BCNF**, если оно находится в **3NF** и в нём отсутствуют зависимости атрибутов первичного ключа от неключевых атрибутов.

Ситуация когда **3NF** не эквивалента **BCNF** возникает, когда имеется несколько АК, которые являются составными и имеют общий атрибут.

Было

- STUDENT_RESEARCH: (NAME, PASSPORT, RESEARCH, ...)
- PK: (NAME, RESEARCH)
- АК: (PASSPORT, RESEARCH)
- ФЗ: PASSPORT \rightarrow NAME
- Аномалия: дублирование данных

Было

- STUDENT_RESEARCH: (NAME, PASSPORT, RESEARCH, ...)
- PK: (NAME, RESEARCH)
- АК: (PASSPORT, RESEARCH)
- ФЗ: PASSPORT \rightarrow NAME
- Аномалия: дублирование данных

Стало

- STUDENTS: (PASSPORT, NAME, ...)
- RESEARCH: (NAME, ...)
- STUDENT_RESEARCH: (STUDENT_PASSPORT, RESEARCH_NAME)

Многозначная зависимость

Разобьём множество атрибутов отношения R на три непересекающиеся группы X , Y и Z . Многозначная зависимость $X \twoheadrightarrow Y$ означает что:

- Если взять любое значение x_c группы атрибутов X , встречающееся в таблице
- Составить список всех комбинаций $x_c y z$, встречающихся в таблице
- Окажется, что x_c всегда ассоциируется с одними и теми же значениями y (независимо от z).

Многозначная зависимость

Разобьём множество атрибутов отношения R на три непересекающиеся группы X , Y и Z . Многозначная зависимость $X \twoheadrightarrow Y$ означает что:

- Если взять любое значение x_c группы атрибутов X , встречающееся в таблице
- Составить список всех комбинаций $x_c yz$, встречающихся в таблице
- Окажется, что x_c всегда ассоциируется с одними и теми же значениями y (независимо от z).

Функциональная зависимость

Функциональная зависимость — это частный случай многозначной зависимости: в функциональной зависимости $X \rightarrow Y$ каждое значение x определяет **в точности одно** значение y .

Четвёртая нормальная форма

Четвёртая нормальная форма (4NF)

Отношение находится в **4NF** в случае, если оно находится в **BCNF** и все многозначные зависимости являются функциональными.

Четвёртая нормальная форма

Четвёртая нормальная форма (4NF)

Отношение находится в **4NF** в случае, если оно находится в **BCNF** и все многозначные зависимости являются функциональными.

Было

- PIZZA_DELIVERY: (RESTAURANT, PIZZA_TYPE, DELIVERY_AREA)
- PK: (RESTAURANT, PIZZA_TYPE, DELIVERY_AREA)
- Многозначные зависимости: RESTAURANT \twoheadrightarrow PIZZA_TYPE, RESTAURANT \twoheadrightarrow DELIVERY_AREA

Четвёртая нормальная форма

Четвёртая нормальная форма (4NF)

Отношение находится в **4NF** в случае, если оно находится в **BCNF** и все многозначные зависимости являются функциональными.

Было

- PIZZA_DELIVERY: (RESTAURANT, PIZZA_TYPE, DELIVERY_AREA)
- PK: (RESTAURANT, PIZZA_TYPE, DELIVERY_AREA)
- Многозначные зависимости: RESTAURANT \twoheadrightarrow PIZZA_TYPE, RESTAURANT \twoheadrightarrow DELIVERY_AREA

Стало

- RESTAURANT_PIZZA: (RESTAURANT, PIZZA_TYPE)
- PK: (RESTAURANT, PIZZA_TYPE)
- RESTAURANT_DELIVERY: (RESTAURANT, DELIVERY_AREA)
- PK: (RESTAURANT, DELIVERY_AREA)

Пятая нормальная форма

Обычно при нормализации производится декомпозиция одного отношения в два. Иногда это сделать не удастся, но возможна декомпозиция в большее число отношений, каждое из которых обладает лучшими свойствами. Такие ФЗ называются **зависимостями по соединению**.

Пятая нормальная форма

Обычно при нормализации производится декомпозиция одного отношения в два. Иногда это сделать не удастся, но возможна декомпозиция в большее число отношений, каждое из которых обладает лучшими свойствами. Такие ФЗ называются **зависимостями по соединению**.

Зависимости по соединению

Наличие **зависимости по соединению** в отношении R означает, что отношение R может быть воссоздано путём соединения множества отношений, каждое из которых содержит подмножество атрибутов R .

Пятая нормальная форма

Обычно при нормализации производится декомпозиция одного отношения в два. Иногда это сделать не удастся, но возможна декомпозиция в большее число отношений, каждое из которых обладает лучшими свойствами. Такие ФЗ называются **зависимостями по соединению**.

Зависимости по соединению

Наличие **зависимости по соединению** в отношении R означает, что отношение R может быть воссоздано путём соединения множества отношений, каждое из которых содержит подмножество атрибутов R .

Пятая нормальная форма (5NF)

Отношение находится в **5NF**, когда оно находится в **4NF** и проведена декомпозиция зависимостей по соединению.

Было

- STUDENT_WORKGROUP_RESEARCH: (STUDENT_ID, WORKGROUP_ID, RESEARCH_ID)

Было

- STUDENT_WORKGROUP_RESEARCH: (STUDENT_ID, WORKGROUP_ID, RESEARCH_ID)

Стало

- STUDENT_WORKGROUP: (STUDENT_ID, WORKGROUP_ID)
- STUDENT_RESEARCH: (STUDENT_ID, RESEARCH_ID)
- WORKGROUP_RESEARCH: (WORKGROUP_ID, RESEARCH_ID)

Было

- ORDERS: (ORDER_ID, CUSTOMER_ID, PRODUCT_ID, PACKAGE_ID)
- ФЗ: ORDER_ID \rightarrow CUSTOMER_ID, ORDER_ID \rightarrow PRODUCT_ID, ORDER_ID \rightarrow PACKAGE_ID

Ещё пример нормализации

Было

- ORDERS: (ORDER_ID, CUSTOMER_ID, PRODUCT_ID, PACKAGE_ID)
- ФЗ: ORDER_ID \rightarrow CUSTOMER_ID, ORDER_ID \rightarrow PRODUCT_ID, ORDER_ID \rightarrow PACKAGE_ID

Один из вариантов

- ORDER_CUSTOMER: (ORDER_ID, CUSTOMER_ID)
- ORDER_PRODUCT: (ORDER_ID, PRODUCT_ID)
- ORDER_PACKAGE: (ORDER_ID, PACKAGE_ID)

И ещё пример нормализации

Было

- SALE: (SALESMAN_ID, BRAND_ID, PRODUCT_TYPE_ID)

И ещё пример нормализации

Было

- SALE: (SALESMAN_ID, BRAND_ID, PRODUCT_TYPE_ID)

Стало

- SALESMAN_PRODUCT_TYPE: (SALESMAN_ID, PRODUCT_TYPE_ID)
- SALESMAN_BRAND: (SALESMAN_ID, BRAND_ID)
- BRAND_PRODUCT_TYPE: (BRAND_ID, PRODUCT_TYPE_ID)

- 3NF считается необходимой и достаточной

- 3NF считается необходимой и достаточной
- Более высокие нормальные формы — при наличии нежелательных аномалий

- 3NF считается необходимой и достаточной
- Более высокие нормальные формы — при наличии нежелательных аномалий
- Денормализация для повышения производительности запросов (некоторые отношения не удовлетворяют 3NF)

- 3NF считается необходимой и достаточной
- Более высокие нормальные формы — при наличии нежелательных аномалий
- Денормализация для повышения производительности запросов (некоторые отношения не удовлетворяют 3NF)
- Существуют и другие нормальные формы