

## **Нормализация**

### **Функциональная зависимость**

- Функциональная зависимость  $A \rightarrow B$  означает, что каждому значению атрибута A соответствует ровно одно значение атрибута B. Говоря проще, зная A, мы однозначно определяем B.
- НомерПаспорта  $\rightarrow$  ФИО (Зная номер паспорта, мы найдем только одного человека).
- НомерЗаказа  $\rightarrow$  ДатаЗаказа, Сумма (Зная номер заказа, мы узнаем его дату и сумму).

# Нормализация

## Первая нормальная форма

- Переменная отношения находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение каждого из атрибутов.

Повторяющиеся группы

ИСУ	Фамилия	Номер 1	Номер 2
1234	Маск	111-11-11	
2452	Торвальд	222-22-22	444-44-44
2132	Гейтс		333-33-33

Неатомарные атрибуты

ИСУ	Фамилии	Номер 1
1234	Маск	111-11-11
2452	Торвальд	222-22-22,444-44-44
2132	Гейтс	

1-ая нормальная форма

ИСУ	Фамилия	Номер
1234	Маск	111-11-11
2452	Торвальд	222-22-22
2452	Торвальд	444-44-44
2132	Гейтс	333-33-33

Отношение находится в первой нормальной форме (1НФ) тогда и только тогда, когда

- в отношении нет повторяющихся групп (атрибутов с одинаковым смыслом)
- все атрибуты атомарны
- у отношения есть ключ

# Нормализация

## Вторая нормальная форма

- Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от каждого её потенциального ключа.

Потенциальный ключ — {ИСУ, Предмет}

ИСУ	Имя	Фамилия	Курс	Предмет	Оценка
1234	Илон	Маск	2	Базы данных	4
1234	Илон	Маск	2	Программирование	5
2452	Линус	Торвальд	2	Базы данных	5
2452	Линус	Торвальд	2	Программирование	5

ИСУ	Имя	Фамилия	Курс
1234	Илон	Маск	2
2452	Линус	Торвальд	2

ИСУ	Предмет	Оценка
1234	Базы данных	5
1234	Программирование	4
2452	Базы данных	5
2452	Программирование	5

## **Нормализация**

### **Третья нормальная форма**

- Переменная отношения находится в третьей нормальной форме, когда она находится во второй нормальной форме, и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых.

Преподаватель	Кафедра	Номер
Иванов	ИС	111-11-11
Петров	ИС	111-11-11
Сидоров	ИКТ	222-22-22

Преподаватель	Кафедра
Иванов	ИС
Петров	ИС
Сидоров	ИКТ

Кафедра	Номер
ИС	111-11-11
ИКТ	222-22-22

## Нормализация

### Нормальная форма Бойса-Кодда

- Переменная отношения находится в НФБК, когда она находится в третьей нормальной форме и ключевые атрибуты составного ключа не зависят от неключевых атрибутов
- Таблицы с простым первичным ключом, находящиеся в ЗНФ, автоматически соответствуют НФБК

Потенциальные ключи: {Номер слота, Время начала}, {Номер слота, Время окончания}, {Лаба, Время начала}, {Лаба, Время окончания}

Номер слота	Время начала	Время окончания	Лаба
1	09:30	10:30	1
1	11:00	12:00	1
1	14:00	15:30	1
2	10:00	11:30	1

Проблема: существует функциональная зависимость Тариф  $\rightarrow$  Номер карта. Детерминант (Тариф) не является потенциальным ключом

## Нормализация

### Многозначные зависимости

- $X$  многозначно определяет  $Y$  тогда и только тогда, когда для каждого значения  $X$  существует набор значений  $Y$ , и этот набор не зависит от значений других атрибутов  $Z$ .
- Если мы зафиксируем значение  $X$ , то получим независимые друг от друга множества соответствующих значений  $Y$  и  $Z$ .
- Многозначная зависимость возникает, когда у одной сущности есть две или более независимых характеристики “многие-ко-многим”.

## **Нормализация**

### **Многозначные зависимости**

- **X** многозначно определяет **Y** тогда и только тогда, когда для каждого значения **X** существует набор значений **Y**, и этот набор не зависит от значений других атрибутов **Z**.
- Если мы зафиксируем значение **X**, то получим независимые друг от друга множества соответствующих значений **Y** и **Z**.
- Многозначная зависимость возникает, когда у одной сущности есть две или более независимых характеристики “многие-ко-многим”.

# Нормализация

## Пример 5НФ

Поставщик	Деталь	Проект
Smith	Болт	A
Smith	Гайка	B
Jones	Болт	A

- Бизнес-правило: "Если поставщик S поставляет деталь P, и деталь P используется в проекте J, и поставщик S работает над проектом J, то поставщик S поставляет деталь P для проекта J".
- Это правило является циклическим ограничением. Оно означает, что наша таблица должна быть закрыта относительно этого правила. Если существуют три пары (S,P), (P,J) и (S,J), то должна существовать и тройка (S,P,J).
- Аномалия: Если мы попробуем декомпозировать эту таблицу на любые две проекции (например, (Поставщик, Деталь) и (Деталь, Проект)), а потом обратно их соединить, мы получим фиктивные (ложные) строки, которых не было в исходном отношении.

## Нормализация

### Пример staff\_property\_inspection

- Отчет о осмотрах объектов недвижимости сотрудниками.
- Проблема: Повторяющиеся группы данных, адрес объекта и имя сотрудника дублируются для каждого осмотра.

property_no	property_address	inspection_date	inspection_time	comments	staff_no	staff_name	car_registration
PG4	6 Lawrence St...	18-Oct-00	10.00	Need to replace	SG37	Ann Beech	M231JGR
PG4	6 Lawrence St...	22-Apr-01	09.00	In good order	SG14	David Ford	M533HDR
PG4	6 Lawrence St...	1-Oct-01	12.00	Damp rot in bathroom	SG14	David Ford	N721HFR

## Нормализация

### 1 НФ

- Устраниены повторяющиеся группы. Первичный ключ: (property\_no, inspection\_date).
- Атрибут property\_address частично зависит только от property\_no, что нарушает 2НФ.

staff\_property\_inspection

property_no	inspection_date	inspection_time	property_address	comments	staff_no	staff_name	car_registration
PG4	18-Oct-00	10.00	6 Lawrence	Need to replace	SG37	Ann Beech	M231JGR
PG4	22-Apr-01	09.00	6 Lawrence	In good order	SG14	David Ford	M533HDR
PG4	1-Oct-01	12.00	6 Lawrence	Damp rot in	SG14	David Ford	N721HFR

## Нормализация НФБК

inspection

property_no	inspection_date	inspection_time	comments	staff_no
PG4	18-Oct-00	10.00	Need to	SG37
PG4	22-Apr-01	09.00	In good	SG14
PG4	1-Oct-01	12.00	Damp rot in	SG14

staff\_car

staff_no	inspection_date	car_registration
SG37	18-Oct-00	M231JGR
SG14	22-Apr-01	M533HDR
SG14	1-Oct-01	N721HFR

## SQL DELETE

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
DELETE FROM [ ONLY ] table_name [*] [ [ AS ] alias ]
[ USING from_item [, ...] ]
[ WHERE condition | WHERE CURRENT OF cursor_name ]
[ RETURNING [ WITH ( { OLD | NEW } AS output_alias [, ...] ) ]
  { * | output_expression [ [ AS ] output_name ] } [, ...] ]
```