



3. Теория проектирования реляционных баз данных

Базы данных

Хисамутдинов М.А.
кафедра №12 НИЯУ МИФИ
2026

План занятия

- 1 Цели проектирования
- 2 Аномалии в данных
- 3 Функциональная зависимость
- 4 Нормализация и нормальные формы
- 5 Денормализация
- 6 Разбор антипаттернов



Цели проектирования

- 1 Соблюдать корректность и целостность данных, т.е. данные в базе данных должны соответствовать реальному миру
 - Заказ не может существовать без клиента
 - Статус заказа не может быть “непонятно”
 - Сумма заказа не может быть отрицательной



Цели проектирования

- 2 Отсутствие избыточности данных, т.е. каждый факт должен храниться в одном месте

Пример факта:

- Телефон клиента
- Курс валюты
- Название категории



Цели проектирования

3 Минимизация аномалий в данных

Аномалии в данных - это нежелательные эффекты при работе с базой данных, которые возникают из-за неудачной структуры таблиц.

Аномалии могут проявляться так:

- нельзя корректно обновить данные
- сложно или опасно данные обновлять
- при удалении теряется важная информация



Пример базы данных

База данных - Поставка товаров

- Информация о поставщике:

Номер поставщика - Sid, Имя поставщика - SName, Место дислокации - City

- Информация о товаре:

Номер товара - Pid, Название - PName, Стоимость - Price

- Информация о поставке:

Какой поставщик поставляет тот или иной товар и в каком
Количестве - Qty

Схема отношения:

ПОСТАВКА ТОВАРОВ(Sid, SName, City, Pid, PName, Price, Qty)



Аномалия вставки

Аномалия вставки - это ситуация, когда невозможно добавить данные об одной сущности, не добавляя данные о другой.

Чтобы сохранить факт А, база данных также требует факт В, который логически не обязателен

Мы хотим добавить нового поставщика, который еще не сделал ни одной поставки товаров, НО

- При добавлении информации о поставке товаров ни Sid, ни Pid не могут быть пропущены.



Аномалия обновления

Аномалия обновления - это ситуация, когда один и тот же логический факт хранится в нескольких строках, и при его изменении не все копии обновляются самостоятельно.

Мы хотим обновить информацию о поставщике, например, его имя,
НО

- Обновив информацию в одном месте, мы получили расхождение с другими строками, которые связаны с данными поставщиками.

Данные начинают противоречить друг другу.



Аномалия удаления

Аномалия удаления - это ситуация, когда удаление одних данных приводит к потере других данных, логически независимых.

Мы хотим удалить информацию о поставке товара, у поставщика, который сделал всего одну поставку, НО

- Удалив информацию о единственной поставке, мы потеряем факт о том, что такой поставщик у нас был.



Как бороться с аномалиями?

- 1** Подумать заранее о том, какие операции над данными будут выполняться
- 2** Нормализация данных (что это такое узнаем сегодня)
- 3** Использовать ограничения базы данных
- 4** Думать о будущем :)



Функциональная зависимость

Пусть задана переменная отношения R , и X и Y являются произвольными подмножествами заголовка R (“составными” атрибутами).

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).

Обозначение: $X \rightarrow Y$.

Транзитивная зависимость: Если выполняется $X \rightarrow Y$ и выполняется $Y \rightarrow Z$, то выполняется также $X \rightarrow Z$. $X \rightarrow Y$ называется функционально, определяющим Y .



Функциональная зависимость

Чтобы обнаружить функциональную зависимость, можно спросить себя:

“Если это поле изменится, что ещё должно измениться вместе с ним?”



Функциональная зависимость

Пациент	Пол	Доктор	Больница
Иван	М	Иванов	Больница №1
Ксения	Ж	Сидоров	Поликлиника №12
Марк	М	Сидоров	Больница №1
Александр	М	Николаев	Больница №3
Елена	Ж	Зайцев	Больница №4

Проверим есть ли следующие функциональные зависимости?

- Пациент → Пол
- Пол → Пациент
- Доктор → Пациент
- Доктор → Больница



Функциональная зависимость

Пациент	Пол	Доктор	Больница
Иван	М	Иванов	Больница №1
Ксения	Ж	Сидоров	Поликлиника №12
Марк	М	Сидоров	Больница №1
Александр	М	Николаев	Больница №3
Елена	Ж	Зайцев	Больница №4

Проверим есть ли следующие функциональные зависимости?

- Пациент → Пол (Да)
- Пол → Пациент (Нет)
- Доктор → Пациент (Нет)
- Доктор → Больница (Нет)



Функциональная зависимость

Госпиталь	Владелец	Возраст
Больница №1	Иван	40
Поликлиника №12	Александр	65
Больница №3	Петр	65

Проверим есть ли следующие функциональные зависимости?

- Госпиталь → Владелец
- Владелец → Возраст
- Госпиталь → Возраст



Функциональная зависимость

Госпиталь	Владелец	Возраст
Больница №1	Иван	40
Поликлиника №12	Александр	65
Больница №3	Петр	65

Проверим есть ли следующие функциональные зависимости?

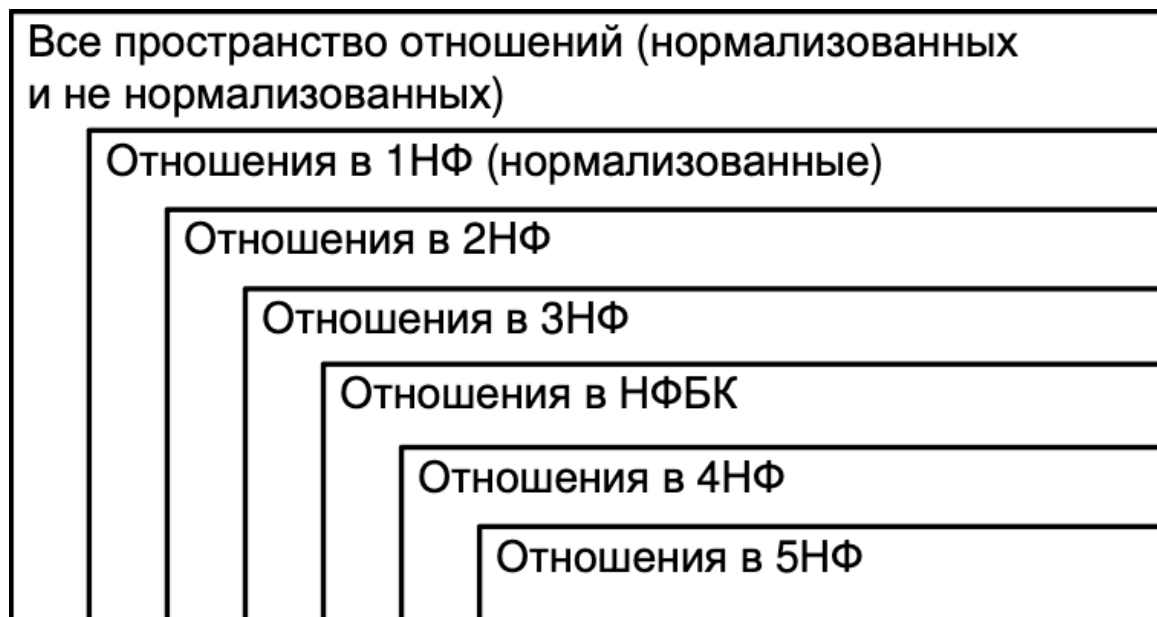
- Госпиталь → Владелец (Да)
- Владелец → Возраст (Да)
- Госпиталь → Возраст (Да, транзитивная)



Нормализация отношений

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами.

Говорят, что отношение находится в некоторой нормальной форме, если оно удовлетворяет заданному набору условий.



Первая нормальная форма (1НФ)

Отношение находится в 1НФ тогда и только тогда, когда все используемые домены содержат только скалярные (атомарные, простые) значения

Пример - ненормализованное отношение

Поставка товаров: PK(Sid: Номер продавца, SName: Имя продавца, City: Город, Code: Код, PS: Поставка)

Составной домен Поставка (Номер, Название, Деньги, Количество)



Первая нормальная форма (1НФ)

Sid	SName	City	Code	PS
S1	Smith	London	20	P1, Rice, 12, 200 P2, Apple, 17, 100 P3, Banana, 17, 100
S2	Jones	Paris	10	P1, Rice, 12, 150 P2, Apple, 17, 200



Первая нормальная форма (1НФ)

Sid	SName	City	Code	Pid	PName	Price	Qty
S1	Smith	London	20	P1	Rice	12	200
S1	Smith	London	20	P2	Apple	17	100
S1	Smith	London	20	P3	Banana	17	100
S2	Jones	Paris	10	P1	Rice	12	150
S2	Jones	Paris	10	P2	Apple	17	200

Главный недостаток 1НФ - избыточность

Проблема обновления

Проблема вставки

Проблема удаления



Вторая нормальная форма (2НФ)

Отношение находится в 2НФ тогда и только тогда, когда оно находится в 1НФ, и каждый его неключевой атрибут **функционально полно** зависит от любого возможного ключа этого отношения

$(Sid, Pid) \rightarrow Qty$ - функционально полная

$(Sid, Pid) \rightarrow Sname$ или $(Sid, Pid) \rightarrow Price$ - не функционально полные, так как есть $Sid \rightarrow Sname$ и $Pid \rightarrow Pname$.

Иначе:

Отношение находится в 2НФ тогда и только тогда, когда оно находится в 1НФ, и каждый его неключевой атрибут полностью зависит от всего ключа, а не от его части



Вторая нормальная форма (2НФ)

Поставщик

<u>Sid</u>	SName	City	Code
S1	Smith	London	20
S2	Jones	Paris	10

Поставляемый товар

<u>Sid</u>	<u>Pid</u>	PName	Price	Qty
S1	P1	Rice	12	200
S1	P2	Apple	17	100
S1	P3	Banana	17	100
S2	P1	Rice	12	150
S2	P2	Apple	17	200



Вторая нормальная форма (2НФ)

Поставщик

<u>Sid</u>	SName	City	Code
S1	Smith	London	20
S2	Jones	Paris	10

Товар

<u>Pid</u>	PName	Price
P1	Rice	12
P2	Apple	17
P3	Banana	17

Поставка

<u>Sid</u>	<u>Pid</u>	Qty
S1	P1	200
S1	P2	100
S1	P3	100
S2	P1	150
S2	P2	200



Третья нормальная форма (3НФ)

Отношение находится в 3НФ тогда и только тогда, когда оно находится в 2НФ, и каждый его неключевой атрибут не транзитивно зависит от ключа

Декомпозиция на основе $City \rightarrow Code$:

Поставщик (Sid, Sname, City)

Город (City, Code)



Нормальная форма Бойса-Кодда

Для случаев когда:

- отношение имеет несколько потенциальных ключей
- потенциальные ключи являются составными
- потенциальные ключи имеют общие атрибуты

Нормализованное отношение находится в НФБК тогда и только тогда, когда каждый детерминант является потенциальным ключом

$f : X \rightarrow Y$; X – детерминант

Отношение **R** находится в НФБК, если оно находится в ЗНФ и все функционально определяющие совокупности атрибутов являются потенциальным ключом.

До определения НФБК, предполагалось, что в отношении один потенциальный ключ, который и является первичным. НФБК может нарушаться в случае, когда потенциальных ключей несколько.



Нормальная форма Бойса-Кодда

В 3НФ допускается:

- функциональная зависимость
- где детерминант **не ключ**
- если зависимое поле - часть ключа

В НФБК:

- никаких исключений
- детерминант всегда ключ



Нормальная форма Бойса-Кодда

Обучение (Student, Course, Teacher):

- каждый студент изучает некоторый предмет под руководством только одного преподавателя $(S, C) \rightarrow T$
- каждый преподаватель проводит занятия только по одному предмету $T \rightarrow C$

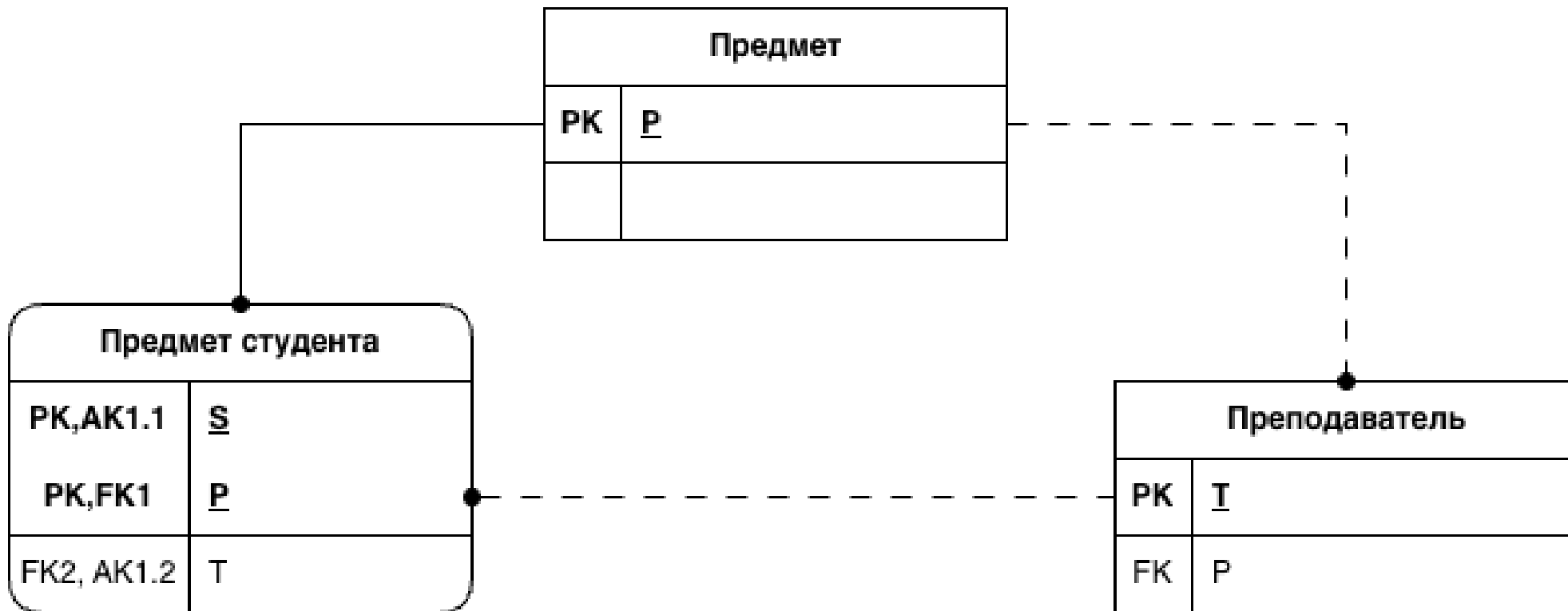
Кандидатные ключи: (Student, Course) и (Student, Teacher)

Student(ПК?)	Course(ПК?)	Teacher(ПК?)
Иванов	Математика	Петров
Иванов	Физика	Михайлов
Сидоров	Математика	Ильин
Сидоров	Физика	Михайлов
Сергеев	Химия	Дымов



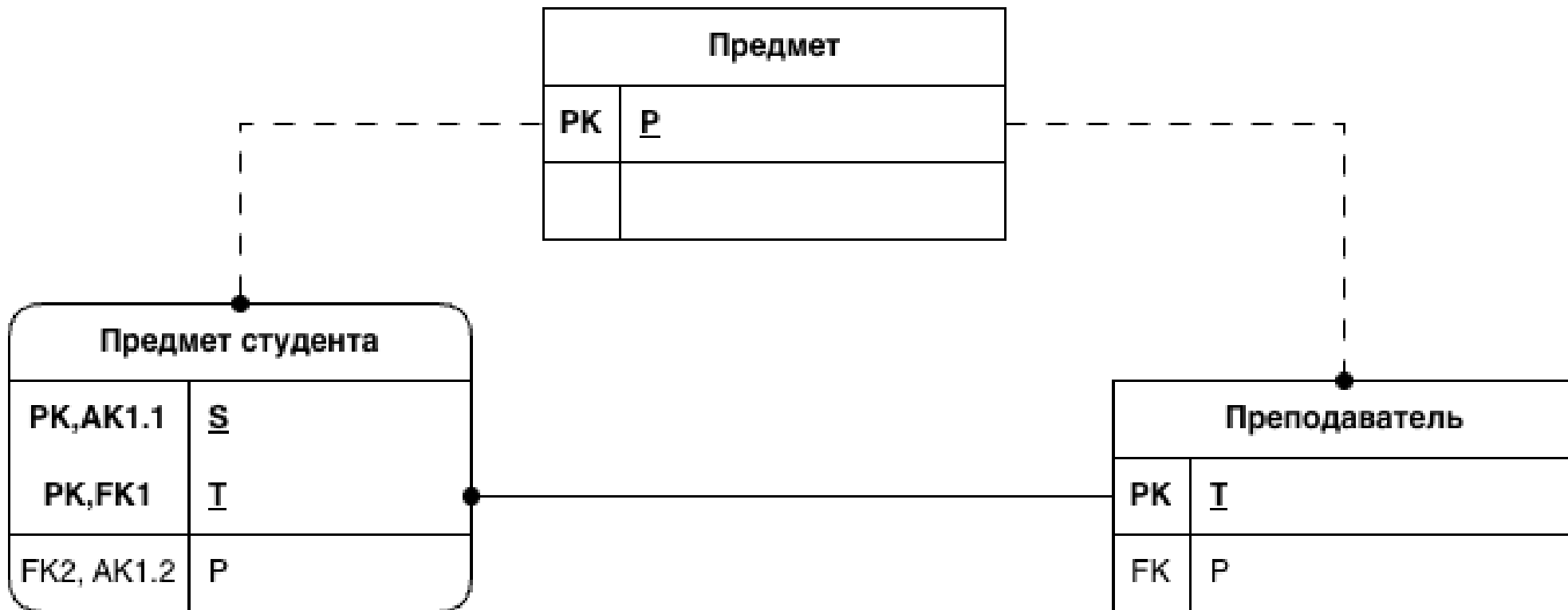
Нормальная форма Бойса-Кодда

Нарушение НФБК, ключ SP



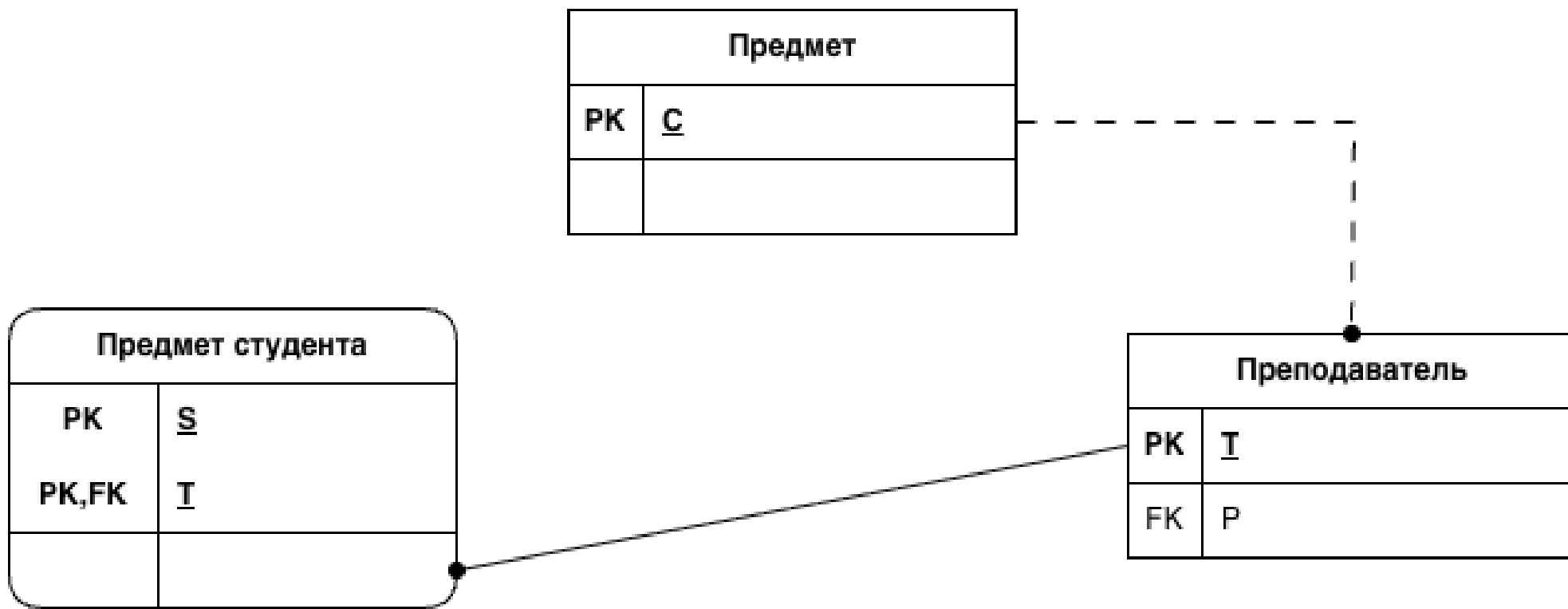
Нормальная форма Бойса-Кодда

Нарушение НФБК, ключ ST

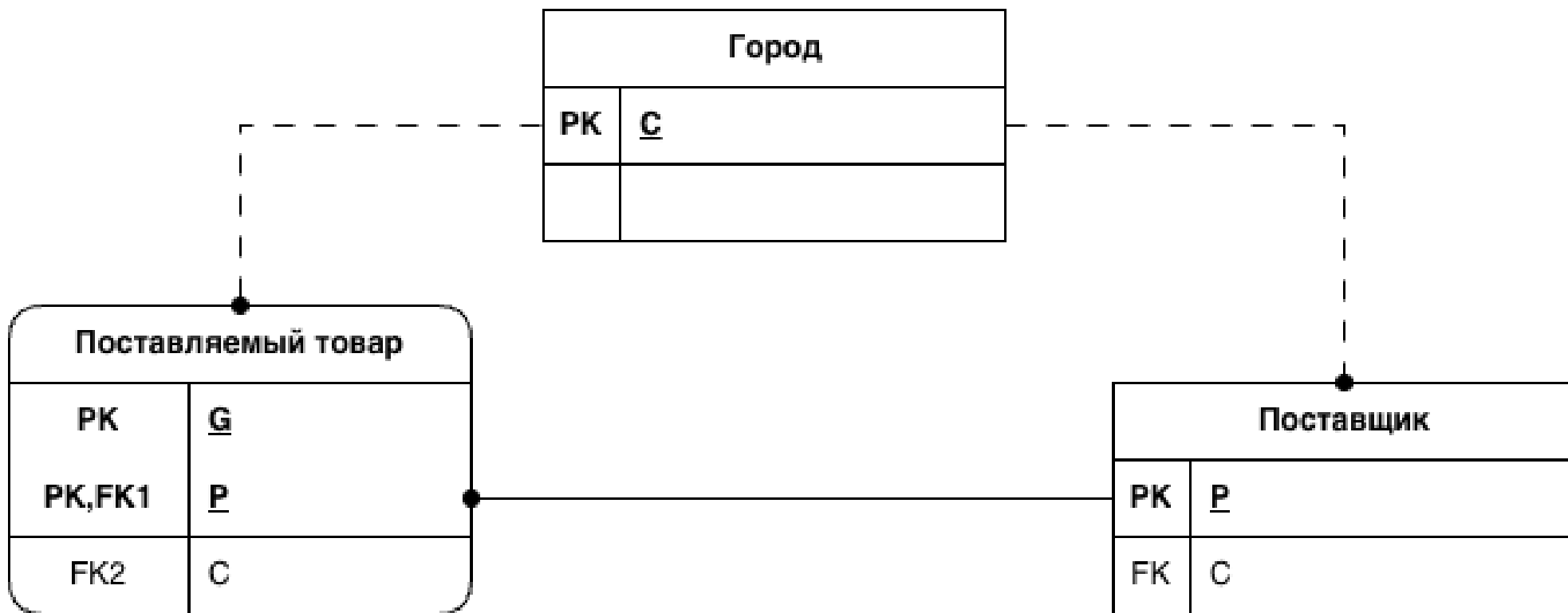


Нормальная форма Бойса-Кодда

Удовлетворяет НФБК



Нормальная форма Бойса-Кодда



Отсутствует функциональная зависимость $(G, C) \rightarrow P$, нет ключа (G, C)



Многозначная зависимость

Ненормализованное отношение

Учебный курс	День занятий	Студент
C1	Понедельник Среда	Иванов Петров Сидоров
C2	Пятница	Быков Мягков



Многозначная зависимость

- Каждый курс может иметь произвольное количество дней занятий
- Каждый курс могут изучать произвольное количество студентов
- Дни занятий и студенты совершенно не зависят друг от друга, т.е. независимо от дня занятий состав группы студентов один и тот же
- Каждый студент может быть связан с любым курсом



Многозначная зависимость

CDS

Учебный курс	День занятий	Студент
C1	Понедельник	Иванов
C1	Понедельник	Петров
C1	Понедельник	Сидоров
C1	Среда	Иванов
C1	Среда	Петров
C1	Среда	Сидоров
C2	Пятница	Быков
C2	Пятница	Мягков

Проблема вставки - добавить день занятий по курсу

Проблема обновления - перенести день занятий по курсу

Проблема удаления - отменить день занятий по курсу



Многозначная зависимость

CD(Учебный курс, День занятий)

CS(Учебный курс, Студент)

CD

Учебный курс	День занятий
C1	Понедельник
C1	Среда
C2	Пятница

CS

Учебный курс	Студент
C1	Иванов
C1	Петров
C1	Сидоров
C2	Мягков
C2	Быков



Многозначная зависимость

Пусть A, B, C - некоторые произвольные подмножества атрибутов схемы отношения $R(A, B, C)$.

Тогда B многозначно зависит от A ($A \twoheadrightarrow B$) тогда и только тогда, когда множество значений B , соответствующее заданной паре $\langle A : a, C : c \rangle$ отношения R зависит только от A , но не зависит от C .

Для данного отношения $R(A, B, C)$ многозначная зависимость $A \twoheadrightarrow B$ выполняется тогда и только тогда, когда также выполняется многозначная зависимость $A \twoheadrightarrow C : A \twoheadrightarrow B | C$

В нашем примере: *Учебный курс* \twoheadrightarrow *День занятий* | *Студент*

Пусть A, B, C - некоторые множества атрибутов схемы отношения $R(A, B, C)$. Отношение R будет равно соединению его проекций $R1(A, B)$ и $R2(A, C)$ тогда и только тогда, когда для отношения выполняется многозначная зависимость $A \twoheadrightarrow B$.



Четвертая нормальная форма (4НФ)

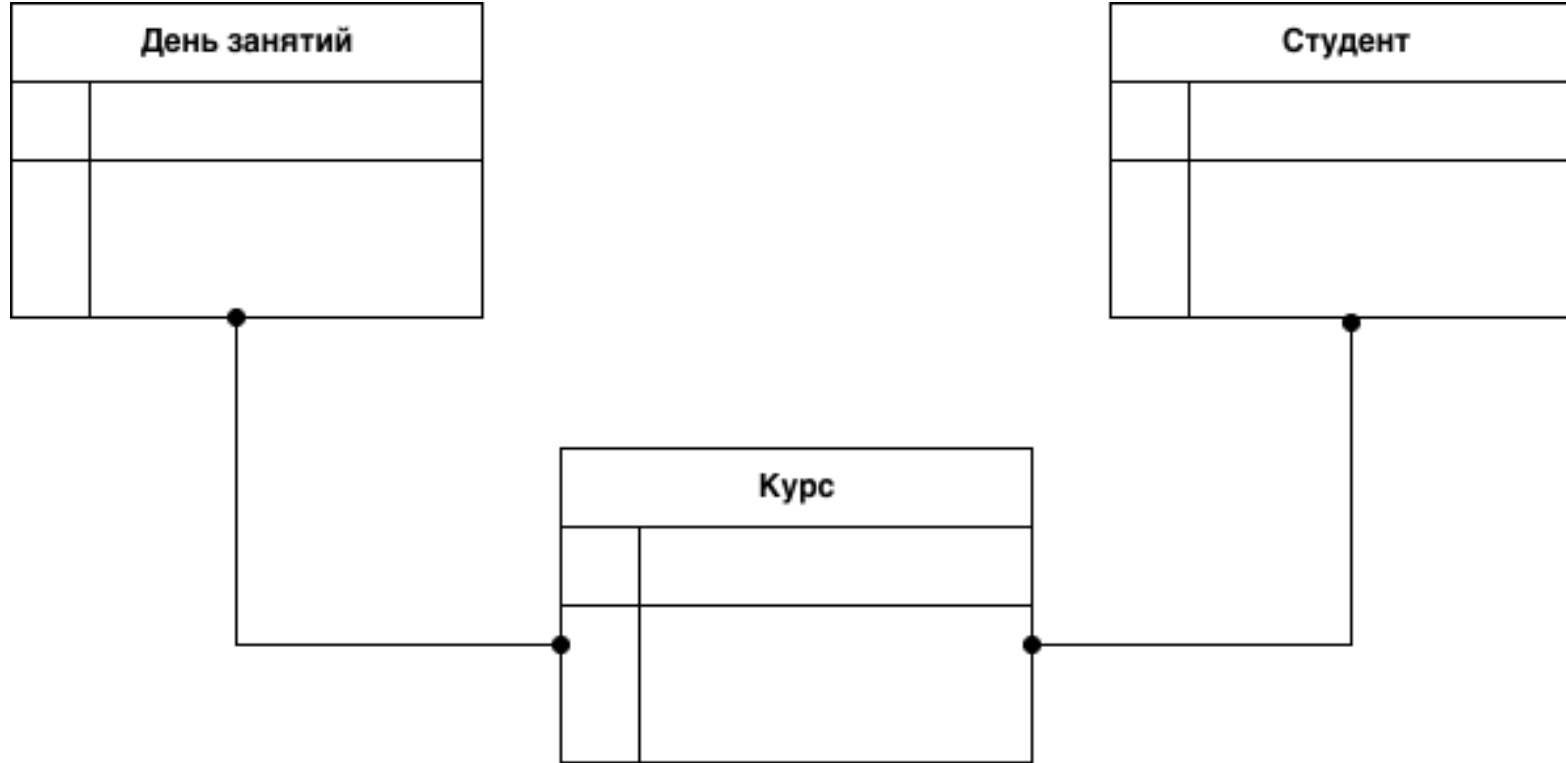
Отношение R находится в 4НФ тогда и только тогда, когда в случае существования многозначной зависимости $A \twoheadrightarrow B$ все атрибуты отношения R функционально зависят от A

(В отношении отсутствуют многозначные зависимости)



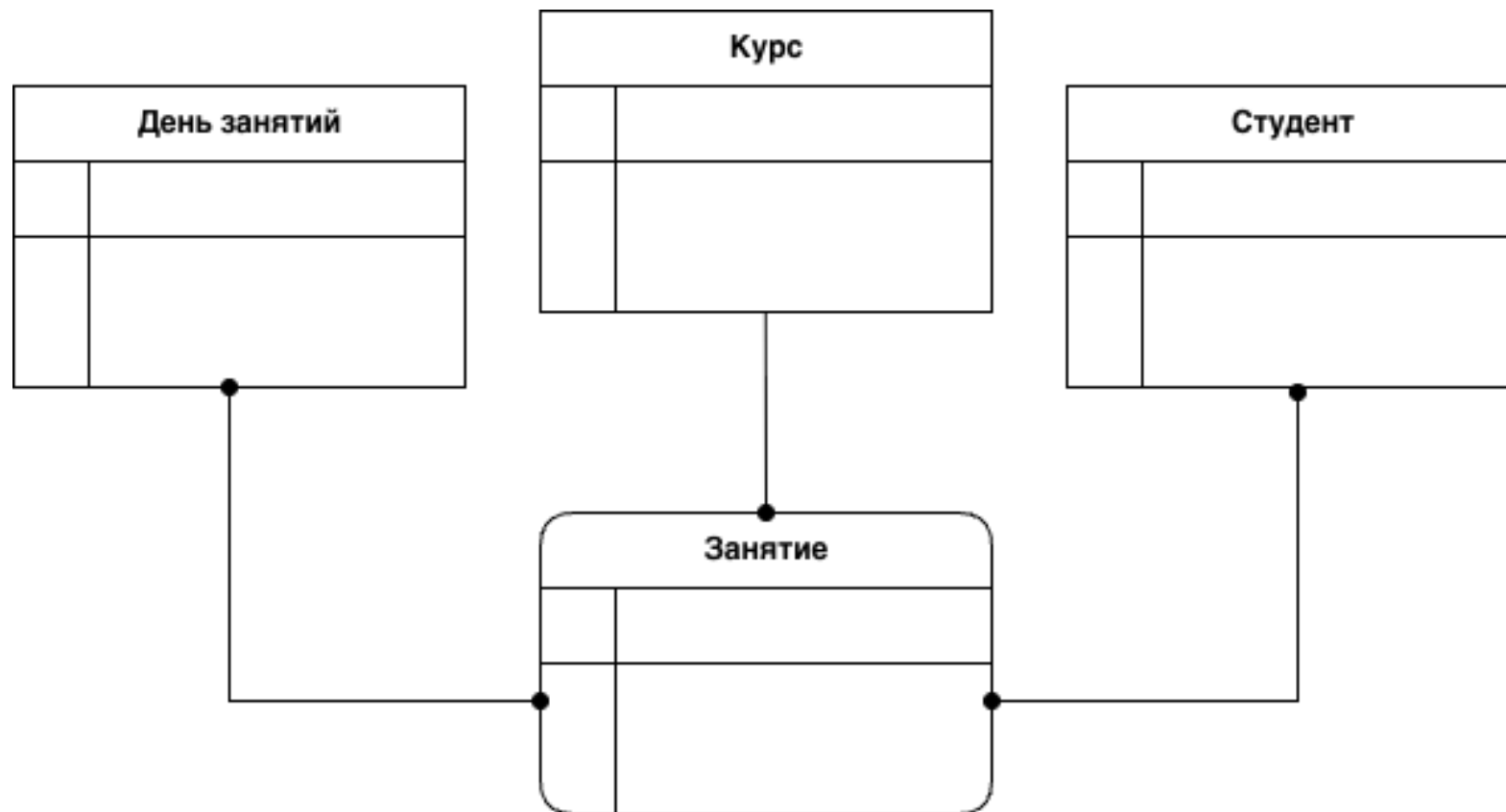
Четвертая нормальная форма (4НФ)

Диаграмма уровня сущностей



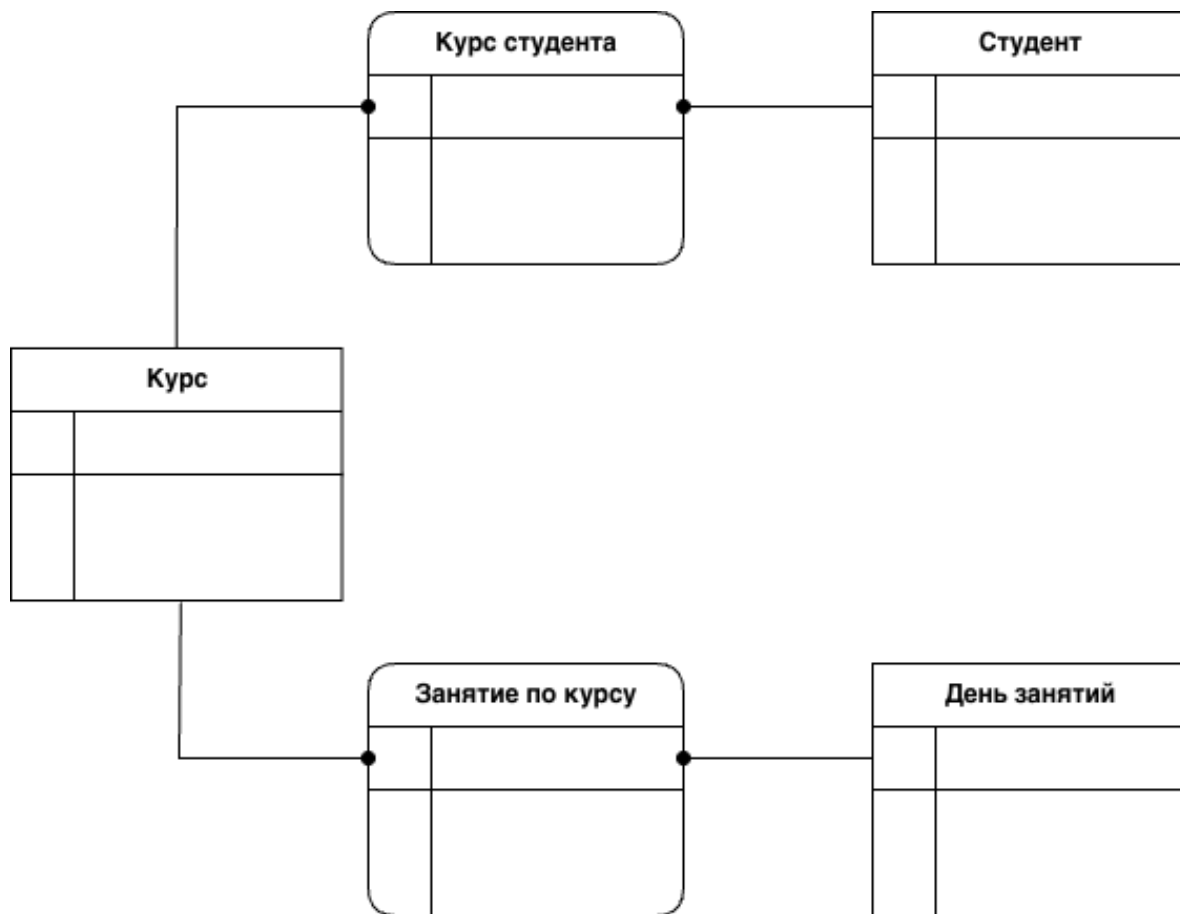
Четвертая нормальная форма (4НФ)

Разрешение неопределенных связей: нарушение 4НФ



Четвертая нормальная форма (4НФ)

Разрешение неопределенных связей: обеспечение требований 4НФ



Пятая нормальная форма (5НФ)

Отношение $R(A_1, A_2, \dots, A_n)$, где A_1, A_2, \dots, A_n - произвольные подмножества атрибутов R , удовлетворяет зависимости соединения $*(A_1, A_2, \dots, A_n)$ тогда и только тогда, когда оно равносильно соединению своих проекций с подмножествами атрибутов A_1, A_2, \dots, A_n (т.е. восстанавливается без потерь путем соединениях своих проекций).

Иными словами:

Таблица находится в 5НФ, если ее нельзя разложить на более простые таблицы без потери информации, и все возможные разложения обусловлены только ключами.



Пятая нормальная форма (5НФ)

Есть: поставщик, товар, склад.

Правила:

- Не каждый поставщик поставляет каждый товар
- Не каждый товар есть на каждом складе
- Поставка возможна только если совпадают все три условия

Поставка(*Поставщик, Товар, Склад*).



Пятая нормальная форма (5НФ)

Поставщик	Товар	Склад
S1	P1	W1
S1	P2	W1
S2	P1	W1

Из парных фактов:

- S1 работает с W1
- P1 есть на W1
- S1 поставляет P1

Но это не означает, что существует кортеж (S2, P2, W1). А наша таблица не может выразить такое ограничение.

Можно привести к 5НФ путем разбиения на минимальные отношения.



Пятая нормальная форма (5НФ)

Товары поставщика

Поставщик	Товар
S1	P1
S1	P2
S2	P1

Склады поставщика

Поставщик	Склад
S1	W1
S2	W1

Товар на складе

Товар	Склад
P1	W1
P2	W1



Денормализация

Это процесс намеренное нарушение нормальных форм (обычно ЗНФ, НФБК) ради производительности, простоты запросов или удобства чтения.

Когда денормализация может быть полезна:

- Read-heavy системы
- Дорогие соединения данных
- Упрощения запросов



Денормализация (пример)

Есть два отношения

Заказы(*ID заказа, ID покупателя, Дата создания*)

Покупатели(*ID покупателя, Имя, Город*)

Проблемы:

- Заказы - миллионы строк
- Покупатели - миллионы
- Много отчетов
- Соединение данных - узкое место

Можно создать денормализованную версию

Факты_заказов(*ID заказа, ID покупателя, Город покупателя, Дата создания*)

Нет соединения и запросы к данным стали проще.



Денормализация (пример)

Какие важные замечания к таблице?

- Источник данных - Покупатели
- Данные **не обновляются** произвольно
- Обновление централизованное (через специальные пайплайны данных)

Когда денормализацию использовать разумно?

- Значимый источник данных
- Из таблицы только читают
- Контролируемое обновление (возникают задержки в появлении данных)

ПЛОХО: “чтобы не писать JOIN”



Антипаттерны нормализации данных

1 Одна таблица для всех данных

Заказы (*ID заказа, ID пользователя, телефон пользователя, товар, категория товара, цена, статус, ...*)

Почему это плохо?



Антипаттерны нормализации данных

1 Одна таблица для всех данных

Заказы (ID заказа, ID пользователя, телефон пользователя, товар, категория товара, цена, статус, ...)

Почему это плохо?

- Все аномалии сразу
- Невозможно поддерживать
- Логика размазана



Антипаттерны нормализации данных

2 Справочники внутри событий

Товары в заказе (*ID заказа, ID товара, название товара, категория товара, цена, количество*)

Почему это плохо?



Антипаттерны нормализации данных

2 Справочники внутри событий

Товары в заказе (*ID заказа, ID товара, название товара, категория товара, цена, количество*)

Почему это плохо?

- Нарушене 2НФ
- Все аномалии



Антипаттерны нормализации данных

3 Злоупотребление составными ключами

Заказ(*ID товара, ID покупателя, количество, цена*)

Почему это может быть проблемой?



Антипаттерны нормализации данных

3 Злоупотребление составными ключами

Заказ(*ID товара, ID покупателя, количество, цена*)

Почему это может быть проблемой?

- Если вы будете ссылаться на данную запись в других таблицах будут составные FK
- Трудные соединения данных

Можно решить проблему с помощью: ID + Unique



Заключение

Цели проектирования: понизить избыточность данных. Повысить надежность и достоверность данных.

Функциональные зависимости: позволяют задать ограничения на этапе проектирования без дополнительных усилий во время разработки.

Нормализация отношений: концепция шести нормальных форм (1НФ, 2НФ, 3НФ, НФБК, 4НФ, 5НФ). Первые три часто используются в реальной жизни, остальные реже.

