

Лекция 9

Аномалии данных. Нормализация базы данных

Аномалии в базах данных

Почему схема БД может быть плохой?

Почему схема БД может быть плохой?

Схема или модель базы данных может быть «плохой» — то есть неэффективной, несогласованной или уязвимой к ошибкам — если она нарушает принципы **нормализации, целостности или семантической корректности**. В результате возникают так называемые **аномалии обновления**, которые приводят к **потере данных, дублированию, противоречивости или ошибкам при работе с БД**.

Почему схема БД может быть плохой?

В теории баз данных **аномалии** — это нежелательные побочные эффекты, возникающие при некачественном проектировании структуры таблиц, особенно когда данные избыточны или ненормализованы. Аномалии приводят к противоречиям, потере данных или ошибкам при обновлении.

Существует три основных типа аномалий:

Аномалии в базе данных

1. Аномалия обновления (модификации)

Проблема: при изменении данных в одной строке приходится обновлять много одинаковых строк, иначе данные станут несогласованными.

Пример:

Таблица **Сотрудники** :

ID	ИМЯ	ОТДЕЛ	РУКОВОДИТЕЛЬ ОТДЕЛА
1	Иван	Продажи	Мария
2	Анна	Продажи	Мария
3	Сергей	IT	Алексей
4	Ольга	Продажи	Мария

👉 Если Марию уволили, и нового руководителя зовут Елена, нужно изменить **все строки с отделом "Продажи"**.

Если забыть обновить одну — получим **противоречие**: у одного сотрудника руководитель — Мария, у других — Елена.

Причина: информация о руководителе отдела дублируется.

2. Аномалия вставки

Проблема: нельзя добавить некоторые данные, не добавив при этом другие (возможно, неизвестные) данные.

Пример:

Та же таблица.

Хотим добавить новый отдел "Маркетинг", но пока в нём нет сотрудников.

Но в текущей структуре нельзя создать запись без сотрудника — ведь каждая строка требует `Имя`.

👉 Мы не можем ввести информацию об отделе, пока не появится хотя бы один сотрудник.

Причина: данные об отделах привязаны к сотрудникам.

Аномалии в базе данных

3. Аномалия удаления

Проблема: при удалении одних данных случайно теряются и другие, важные данные.

Пример:

Уволился последний сотрудник из отдела IT — Сергей.

Удаляем его строку:

ID	ИМЯ	ОТДЕЛ	РУКОВОДИТЕЛЬ ОТДЕЛА
3	Сергей	IT	Алексей

👉 После удаления исчезает вся информация об отделе IT и его руководителе Алексее, даже если отдел временно пуст!

Причина: данные об отделе хранятся только через сотрудников.

Как избежать аномалий?

С помощью нормализации — процесса приведения структуры БД к нормальным формам.

В нашем примере достаточно разделить таблицу на две:

Нормализация базы данных

Отделы:

ОТДЕЛ	РУКОВОДИТЕЛЬ
Продажи	Мария
IT	Алексей
Маркетинг	?

Сотрудники:

ID	ИМЯ	ОТДЕЛ
1	Иван	Продажи
2	Анна	Продажи
3	Сергей	IT

Нормализация базы данных

Теперь:

- Руководителя можно обновить в одном месте → нет аномалии обновления.
 - Отдел можно добавить без сотрудников → нет аномалии вставки.
 - Увольнение сотрудника не удаляет информацию об отделе → нет аномалии удаления.
-

Вывод

Аномалии — это симптомы плохого проектирования БД.

Их устранение — одна из главных целей нормализации, которая помогает:

- устранить дублирование,
- обеспечить целостность данных,
- сделать БД гибкой и безопасной.

Нормализация реляционной базы данных

Нормализация базы данных — это процесс приведения структуры реляционной базы данных к набору «нормальных форм» с целью:

- устранения избыточности данных,
- предотвращения аномалий обновления, вставки и удаления,
- обеспечения логической целостности и гибкости схемы.

Это делается путём **декомпозиции (разбиения)** таблиц на более мелкие, связанные между собой через внешние ключи, при сохранении **эквивалентности данных** (без потерь информации).

Нормализация базы данных

Таким образом, **нормализация** помогает адекватно выбрать:

- какие сущности действительно независимы,
- какие атрибуты логически принадлежат той или иной сущности,
- как правильно связать сущности (например, через внешние ключи).

Важное уточнение

Хотя нормализация **направляет и проверяет** адекватность выбора сущностей и атрибутов, **сам процесс выявления сущностей и их связей начинается ещё на этапе концептуального моделирования** (например, через ER-диаграммы), **до** формальной нормализации. Нормализация — это скорее **формальный инструмент проверки и уточнения** этих решений на логическом уровне.

Дублирование и избыточность данных

Книга	Студент
Книга 1	Иванов
Книга 2	Петров
Книга 3	Иванов

Данные
дублированные,
но не избыточные

Книга	Студент	Телефон
Книга 1	Иванов	9631440779
Книга 2	Петров	9170550011
Книга 3	Иванов	9631440779

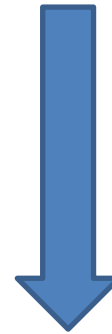
Данные по телефонам
избыточные

В чем опасность избыточности данных в БД?

Дублирование и избыточность данных

Книга	Студент	Телефон
Книга 1	Иванов	9631440779
Книга 2	Петров	9170550011
Книга 3	Иванов	9631440779

Данные по телефонам
избыточные



Разбиваем на две таблицы:

Книга	Студент
Книга 1	Иванов
Книга 2	Петров
Книга 3	Иванов

Студент	Телефон
Иванов	9631440779
Петров	9170550011

Нормальные формы

Первая НФ: атомарные атрибуты.

Требует, чтобы каждое поле таблицы:

- было неделимым (не должно делиться на более мелкие значения)
- не содержало повторяющихся групп или массивов.

Вторая НФ: отсутствие зависимостей частичного ключа.

Все поля таблицы должны зависеть от первичного ключа, т.е. первичный ключ однозначно определяет запись и не является избыточным.

Те поля, которые зависят только от части первичного ключа, должны быть выделены в отдельные таблицы.

Нормальные формы

Третья НФ: **устранение транзитивных зависимостей.**

В таблице не должно быть транзитивных зависимостей между неключевыми полями, т.е. чтобы значение любого поля таблицы, не входящего в первичный ключ, не зависело от значения другого поля, не входящего в первичный ключ.

Первая НФ

Студент	Телефон
Иванов	9631440779; 9780231200
Петров	9170550011; 9051231234

Первая НФ

Студент	Телефон
Иванов	9631440779; 9780231200
Петров	9170550011; 9051231234

Есть повторяющиеся группы –
нарушение 1 НФ

Первая НФ

Студент	Телефон
Иванов	9631440779; 9780231200
Петров	9170550011; 9051231234

Есть повторяющиеся группы –
нарушение 1 НФ

Студент	Телефон
Иванов	9631440779
Иванов	9780231200
Петров	9051231234
Петров	9170550011

Таблица приведена к 1 НФ

Пример

Спроектировать БД для автоматизации отпуска товаров со склада

Накладная № 5

Дата: 03.12.2016 Покупатель: ООО "Друг" Адрес: г. Саранск

Наименование	Кол-во	Ед. изм.	Цена	Сумма
Тушенка	1000	Банка	50	50000
Сахар	100	Кг	30	3000

Итого: 53000

Одна таблица «Отпуск_товара»

Номер	Дата	Покупатель	Адрес	Наименование	Ед_изм	Цена	Кол-во	Сумма
5	03.12.2016	ООО Друг	Саранск	Тушенка	Банка	50	1000	50000
5	03.12.2016	ООО Друг	Саранск	Сахар	Кг	30	100	3000

Отпуск_товаров
Номер
Дата
Покупатель
Адрес
Наименование
Ед_изм
Цена
Количество
Сумма

Одна таблица «Отпуск_товара»

Номер	Дата	Покупатель	Адрес	Наименование	Ед_изм	Цена	Кол-во	Сумма
5	03.12.2016	ООО Друг	Саранск	Тушенка	Банка	50	1000	50000
5	03.12.2016	ООО Друг	Саранск	Сахар	Кг	30	100	3000

Отпуск_товаров
Номер
Дата
Покупатель
Адрес
Наименование
Ед_изм
Цена
Количество
Сумма

Таблица приведена к 1 НФ ?

Одна таблица «Отпуск_товара»

Номер	Дата	Покупатель	Адрес	Наименование	Ед_изм	Цена	Кол-во	Сумма
5	03.12.2016	ООО Друг	Саранск	Тушенка	Банка	50	1000	50000
5	03.12.2016	ООО Друг	Саранск	Сахар	Кг	30	100	3000

Отпуск_товаров
Номер
Дата
Покупатель
Адрес
Наименование
Ед_изм
Цена
Количество
Сумма

Таблица приведена к 1 НФ ? Да

Первичный ключ?

Выделение таблицы «Товары»

Отпуск_товаров
Номер
Наименование
Дата
Покупатель
Адрес
Ед_изм
Цена
Количество
Сумма

← Первичный ключ

Какие поля зависят только от части
первичного ключа?

Выделение таблицы «Товары»

Отпуск_товаров
Номер
Наименование
Дата
Покупатель
Адрес
Ед_изм
Цена
Количество
Сумма

← Первичный ключ

«Ед_изм» и «Цена» зависят только от «Наименование»

Выделение таблицы «Товары»

Отпуск_товаров
Номер
Наименование
Дата
Покупатель
Адрес
Ед_изм
Цена
Количество
Сумма

← Первичный ключ

«Ед_изм» и «Цена» зависят только от «Наименование»

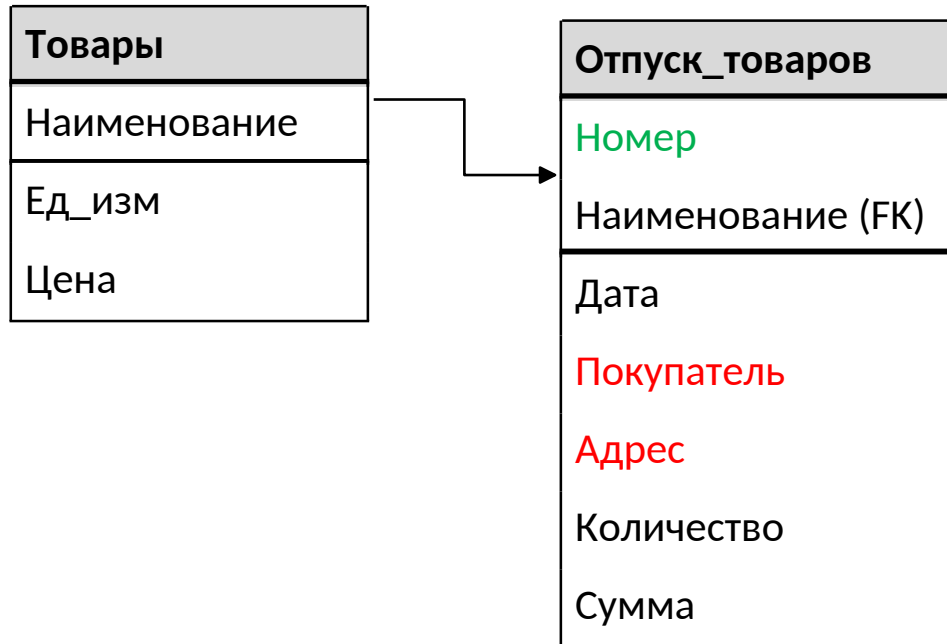


Товары
Наименование
Ед_изм
Цена

Отпуск_товаров
Номер
Наименование (FK)
Дата
Покупатель
Адрес
Количество
Сумма

Выделение таблицы «Покупатели»

«Покупатель» и «Адрес» зависят только от номера накладной



Выделение таблицы «Покупатели»

«Покупатель» и «Адрес» зависят только от номера накладной



Товары
Наименование
Ед_изм
Цена

Отпуск_товаров
Номер
Наименование (FK)
Дата
Количество
Сумма

Покупатели
Покупатель
Адрес

Выделение таблицы «Накладные»

«Дата» зависит только от номера накладной

Товары
Наименование
Ед_изм
Цена

Отпуск_товаров
Номер
Наименование (FK)
Дата
Количество
Сумма

Покупатели
Покупатель
Адрес

Выделение таблицы «Накладные»

«Дата» зависит только от номера накладной



Товары
Наименование
Ед_изм
Цена

Отпуск_товаров
Номер
Наименование (FK)
Количество
Сумма

Покупатели
Покупатель
Адрес

Накладные
Номер
Дата

Устанавливаем связи



Приведена ли БД к 2-й НФ?

Устанавливаем связи



БД приведена к 2-й НФ

Устанавливаем связи



Приведена ли БД к 3-й НФ?

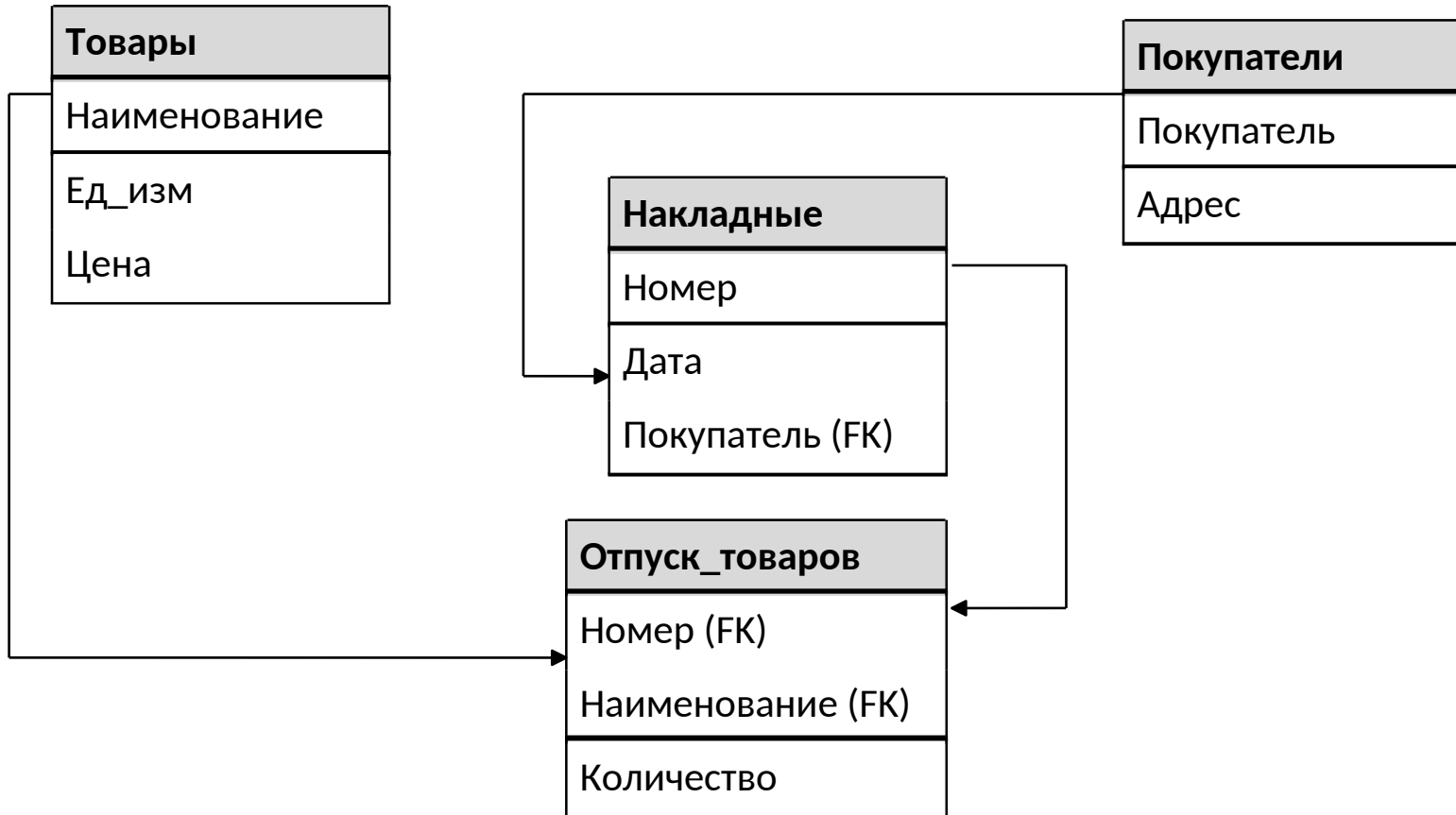
Устанавливаем связи



«Сумма» зависит от «количества»

Удаляем лишние поля

"Отпуск_товаров"."Сумма"="Отпуск_товаров"."Количество"*"Товары"."Цена"



БД приведена к 3-й НФ

Основные нормальные формы

1. 1НФ (Первая нормальная форма)

— Все атрибуты атомарны (неделимы), нет повторяющихся групп или массивов в ячейках.

2. 2НФ (Вторая нормальная форма)

— Таблица в 1НФ, и все неключевые атрибуты полностью зависят от всего первичного ключа (а не от его части).

3. 3НФ (Третья нормальная форма)

— Таблица в 2НФ, и нет транзитивных зависимостей: неключевые атрибуты не зависят от других неключевых.

4. НФБК (Нормальная форма Бойса–Кодда)

— Усиленная версия 3НФ: каждая нетривиальная функциональная зависимость должна зависеть от суперключа.

5. 4НФ, 5НФ и далее

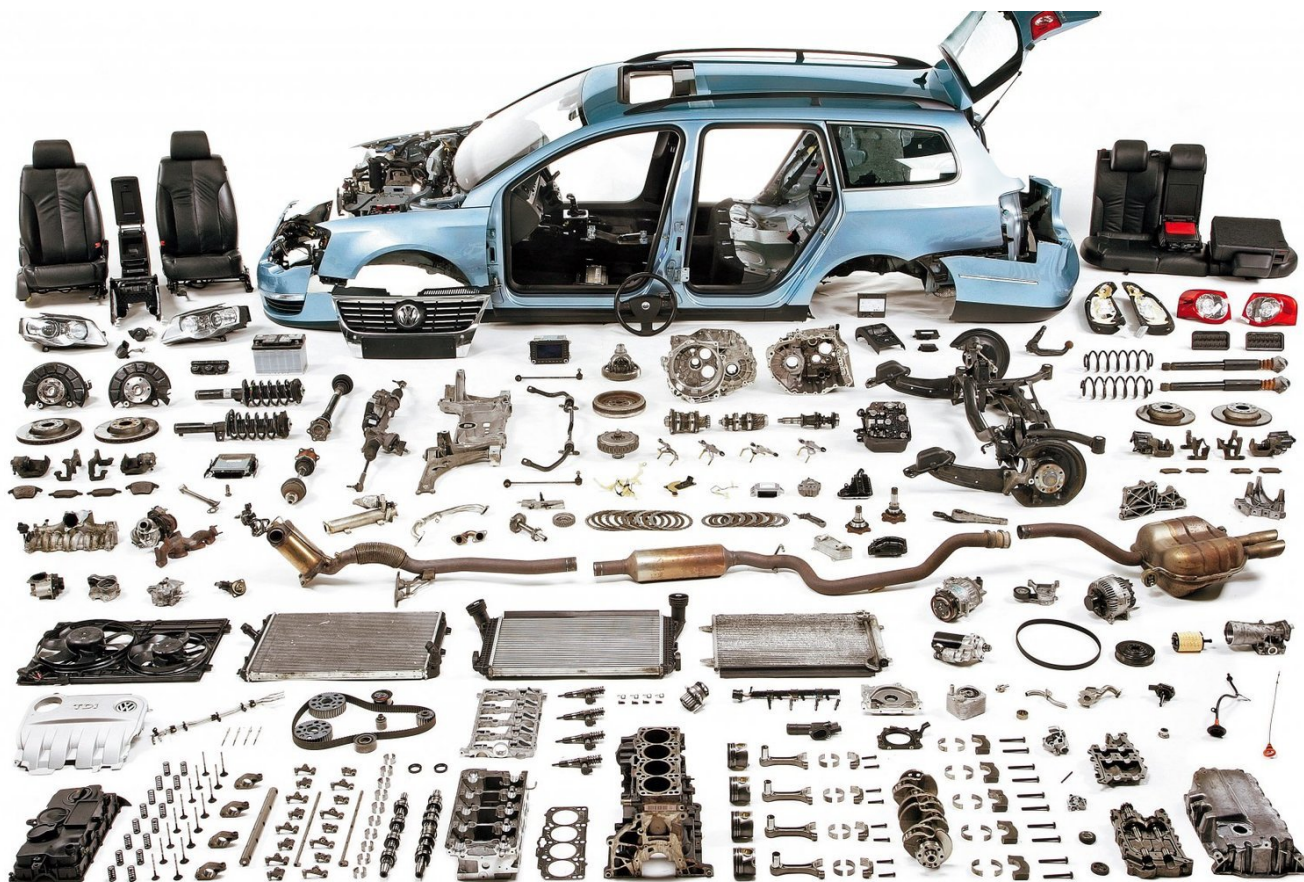
— Устраняют зависимости, связанные с многозначными и соединительными зависимостями (реже используются на практике).

Результаты нормализации

Плюс. Таблицы содержат только один элемент избыточных данных – это поле связи, присутствующее одновременно у родительской и дочерних таблиц.

Минус. Сильно увеличивается число таблиц:

- Теряется восприятие системы в целом.
- Усложняются запросы, приходится объединять данные из разных таблиц (скорость выполнения падает)



Нормализованный автомобиль