

ERD e-commerce

Нормализация реляционных баз данных

Искусство проектирования "хороших" схем:
от избыточности к порядку

🔍 Конструирование Баз данных - Программная инженерия ФКН ВШЭ



Лекция 4



2026

1

Проблема: Аномалии данных

Что такое "плохая" схема? Знакомство с аномалиями

2

Диагностический инструмент

Концепция функциональной зависимости (ФЗ)

3

Пошаговое лечение: Нормальные формы

1НФ • 2НФ • 3НФ • НФБК (BCNF)

4

Прагматика: Денормализация

Когда она оправдана и как применять

⚠ Проблема: "Кошмарная таблица"

🏠 UnnormalizedOrders

Все данные в одном месте — причина проблем

OrderID

ProductID

OrderDate

CustomerID

CustomerName

ProductDescription

ProductPrice

Quantity

🔑 **Первичный ключ**
(OrderID, ProductID) — составной ключ

Три типа аномалий данных



Аномалия вставки

Нельзя добавить новый товар без существующего заказа



Аномалия удаления

Удаление заказа приводит к потере информации о клиенте



Аномалия обновления

Избыточность данных требует обновления в нескольких местах

Функциональная зависимость (ФЗ)

Определение

Атрибут **B** функционально зависит от атрибута **A**

A → B

Если каждому значению **A** соответствует в точности одно значение **B**

Примеры ФЗ

OrderID → OrderDate

CustomerID → CustomerName

ProductID → ProductDescription, ProductPrice

(OrderID, ProductID) → Quantity

Σ Аксиомы Армстронга

💡 Три базовые аксиомы

1

Рефлексивность

Если $B \subseteq A$, то $A \rightarrow B$

2

Дополнение (Augmentation)

Если $A \rightarrow B$, то $AC \rightarrow BC$

3

Транзитивность

Если $A \rightarrow B$ и $B \rightarrow C$, то $A \rightarrow C$

⚙️ Доп. правила

Декомпозиция

Если $A \rightarrow BC$, то $A \rightarrow B$ и $A \rightarrow C$

Объединение

Если $A \rightarrow B$ и $A \rightarrow C$, то $A \rightarrow BC$



Замыкание атрибутов и поиск ключей



Замыкание множества атрибутов

X^+

Замыкание X^+ множества атрибутов X :
это множество всех атрибутов,
которые функционально зависят от X

(до которых можно дойти по стрелочкам)

Алгоритм вычисления замыкания

- 1 Накапливаем **result = X**
- 2 Добавляем **B** , если **$A \subseteq \text{result}$**
- 3 Повторяем, пока **result** не стабилизируется

Поиск ключей

Пример отношения

$R(A, B, C, D, E)$

ФЗ: $A \rightarrow B$

$BC \rightarrow D$

$E \rightarrow A$

Потенциальный ключ: **(A, C, E)**

Суперключ

X^+ содержит все атрибуты

Потенциальный ключ

Суперключ без избыточности

1 Первая нормальная форма (1НФ)

Цель

- ✓ Устранить хаос
- ✓ Обеспечить атомарность

Требование

Все атрибуты должны быть **атомарными**

В каждой ячейке только одно значение.
Никаких списков, массивов, JSON-объектов.

💡 Атомарность

Атомарность зависит от **бизнес-требований**

✓ Атомарно

- Телефон клиента
- Город
- Адрес доставки
- Имя клиента

←?→

✗ Неатомарно

- Список телефонов
- JSON-объект
- Адрес доставки
- Массив данных

❓ Вопрос: нужно ли искать по части атрибута?

❗ Пример нарушения 1НФ

☰ Таблица Clients

ClientID	Name	Phones
101	Иван Иванов	+79261112233, +74955556677

🔍 **Нельзя сравнить список со скаляр** (`client.phones = delivery.phone`)
В современных системах можно, но 50 лет назад это спасло от проблем

🛡️ **Нарушение целостности**
Невозможно обеспечить уникальность (один телефон только у одного клиента)

🔄 **Сложное изменение**
Необходимо парсить и собирать строки

2 Вторая нормальная форма (2НФ)

Цель

Устранить зависимости от
части составного ключа

Требование

Каждый неключевой атрибут **полностью функционально зависит** от первичного ключа

Если ключ составной, все остальные столбцы должны зависеть от **всего ключа целиком**, а не от его части

🔍 Частичные зависимости

🔑 Составной ключ: **(OrderID, ProductID)**

→ **OrderID** → OrderDate, CustomerID

- Зависят только от **OrderID** (части ключа)

→ **ProductID** → ProductDescription, ProductPrice

- Зависят только от **ProductID** (части ключа)

✓ **(OrderID, ProductID)** → Quantity

↗ Декомпозиция к 2НФ

✓ Результат декомпозиции

1 Orders

OrderID (PK)

OrderDate

CustomerID

CustomerName

2 Products

ProductID (PK)

ProductDescription

ProductPrice

3 OrderItems

OrderID (FK)

ProductID (FK)

Quantity

⚙ Устранены частичные зависимости

3 Третья нормальная форма (3НФ)

Цель

Устранить зависимости между
неключевыми атрибутами

Требование

Отсутствие **транзитивных функциональных зависимостей**

Неключевой столбец **не должен зависеть**
от другого **неключевого** столбца

🔍 Транзитивные зависимости

📊 Таблица **Orders**

→ **OrderID** → **CustomerID**

- ФЗ от ключа (правильно)

→ **CustomerID** → CustomerName

- ФЗ между неключевыми атрибутами

⚠ Транзитивная зависимость

OrderID → CustomerID → CustomerName

Декомпозиция к 3НФ

Результат после 3НФ

1

Customers

CustomerID (PK)

CustomerName

2

Products

ProductID (PK) ProductDesc

ProductPrice

3

Orders

OrderID (PK) OrderDate

CustomerID (FK)

4

OrderItems

OrderID (FK) ProductID (FK)

Quantity

Мнемоника для запоминания

Каждый неключевой атрибут
должен зависеть ...

1НФ

от ключа

2НФ

от ключа целиком

3НФ

ни от чего, кроме ключа

✓ Нормальная форма Бойса-Кодда (НФБК, BCNF)

🛡️ Усиленная версия 3НФ

Требование

В **каждой** нетривиальной функциональной зависимости $X \rightarrow Y$, детерминант (слева от стрелки) X является **суперключом**

📌 Замечания

В **99% практических случаев** хорошо спроектированная схема в 3НФ также окажется и в НФБК

НФБК декомпозирует таблицы **без гарантии сохранения** всех функциональных зависимостей

🔍 Пример нарушения НФБК

🏢 Enrollments(Student, Course, Professor)

Потенциальные ключи

- (Student, Course)
- (Student, Professor)

Функциональные зависимости

- ✓ (Student, Course) \rightarrow Professor
- ✓ Professor \rightarrow Course

⚠️ Проблема

Professor \rightarrow Course нарушает НФБК, так как Professor - не суперключ



Алгоритм синтеза (схемы в 3НФ)



Входные данные

▣ Множество атрибутов

Σ Множество ФЗ

❗ Зачем нужен?

Автоматически строит схему в 3НФ



5 шагов синтеза

1

Найти **минимальное покрытие** множества ФЗ

2

Создать таблицу для каждой ФЗ: **A → B** → таблица **(A, B)**

3

Сгруппировать таблицы с одинаковой левой частью

4

Проверить, содержит ли таблица **потенциальный ключ**
Если нет, создать еще одну таблицу с ключом

Пример синтеза схемы

Исходные данные

Атрибуты

(StudentID, StudentName, CourseID, CourseName, Grade)

Функциональные зависимости

- 1 StudentID \rightarrow StudentName
- 2 CourseID \rightarrow CourseName
- 3 (StudentID, CourseID) \rightarrow Grade

✓ Результат: 3 таблицы в 3НФ

1 Students

StudentID (PK)

StudentName

2 Courses

CourseID (PK)

CourseName

3 Grades

StudentID (FK)


CourseID (FK)

Grade




🛡️ **Гарантированно** сохраняет все исходные зависимости

Де-нормализация

Осознанное решение

 Добавление избыточных данных для **повышения производительности**

Когда оправдано?

-  Отчеты и аналитика (**OLAP**)
-  Историчность данных
-  Кэширование

! Важное условие
Только после **нормализованной** схемы и **реальных проблем** с производительностью

Итоги лекции

Нормализация

🔑 Формальный процесс декомпозиции **плохих** таблиц на **хорошие**

×
Избыточность

⚠️
Аномалии

Σ Диагностика

Функциональные зависимости (**ФЗ**)

Нормальные формы

1

Первая нормальная форма (1НФ)

Атомарность значений

2

Вторая нормальная форма (2НФ)

Без частичных зависимостей

3

Третья нормальная форма (3НФ)

Без транзитивных зависимостей

НФБК

Нормальная форма Бойса-Кодда

Строгая версия 3НФ без сохранения зависимостей

Денормализация после нормализации для ускорения