

# KARPOV.COURSES >>> КОНСПЕКТ



## > Конспект > 2 урок > Нормальные формы: Практикум

### > Оглавление

> Оглавление

> Первая итерация нормализации таблицы

Проверка на 2НФ

Достижение 3НФ

> Вторая итерация нормализации таблицы

Несоответствие 2НФ

Приведение к 3НФ

> Третья итерация нормализации таблицы

Приведение к 2НФ

Приведение к 3НФ

Приведение к 4НФ

> Итоги

## > Первая итерация нормализации таблицы

Рассмотрим следующую таблицу с данными о преподавателях, которая похожа на то, как все устроено в Karrov.Courses.

Table
Название курса
Тип курса
Описание курса
Подготовка
Длительность
ФИО хедлайнера
Биография хедлайнера
Наименование блока
Описание блока
ФИО преподавателя
Биография преподавателя

Файл таблицы с данными можно скачать [тут](#).

В первую очередь стоит проверить эту таблицу на первую нормальную форму – повторяются ли какие-то строки, есть ли неатомарные данные.

- Единственный атрибут, где нет повторяющихся строк – это **Наименование блока**.
- Все атрибуты, кроме одного – атомарны. В колонке **Наименование блока** у всех значений есть как название блока, так и его номер. Если в теории нам когда-то понадобятся имя блока и его номер, то мы встретимся с проблемой неатомарности этого атрибута. Тогда придется разбить этот атрибут на два – **Наименование блока** и **Номер блока**.

Попробуем выбрать первичный ключ для таблицы. Как мы помним, ключ должен отвечать двум требованиям:

- Уникальность – значения ключа не должны повторяться.
- Неизбыточность (в случае составного ключа) – никакое подмножество ключа не должно быть уникальным.

Большая часть данных повторяется. Единственный уникальный атрибут – **Наименование блока**. Выберем его в качестве первичного ключа и отметим в схеме.

Table
Название курса
Тип курса
Описание курса
Подготовка
Длительность
ФИО хедлайнера
Биография хедлайнера
<b>Наименование блока</b>
Описание блока
ФИО преподавателя
Биография преподавателя

Таблица с выбранным первичным ключом

Так как ключ простой, то свойство избыточности выполняется автоматически.

## Проверка на 2НФ

Так как выбранный нами ключ – простой, то все остальные функциональные зависимости относятся к этому ключу. Следовательно, таблица находится во второй нормальной форме.

## Достижение 3НФ

Теперь необходимо разобраться с транзитивными зависимостями.

- **Биография преподавателя** зависит от **ФИО преподавателя**.
- **ФИО преподавателя** зависит от **Наименования блока**.
- **Биография хедлайнера** зависит от **ФИО хедлайнера**.
- С колонками, относящимися к курсу все не так просто. **Название курса**, **Описание курса** и **Подготовка** могут являться ключом, если исходить из

таблицы с данными. Мы же выберем в качестве первичного ключа **Название курса**, так как это самый логичный вариант, если исходить не от данных, а от предмета данных.

Для разрешения этих зависимостей введем новые таблицы.

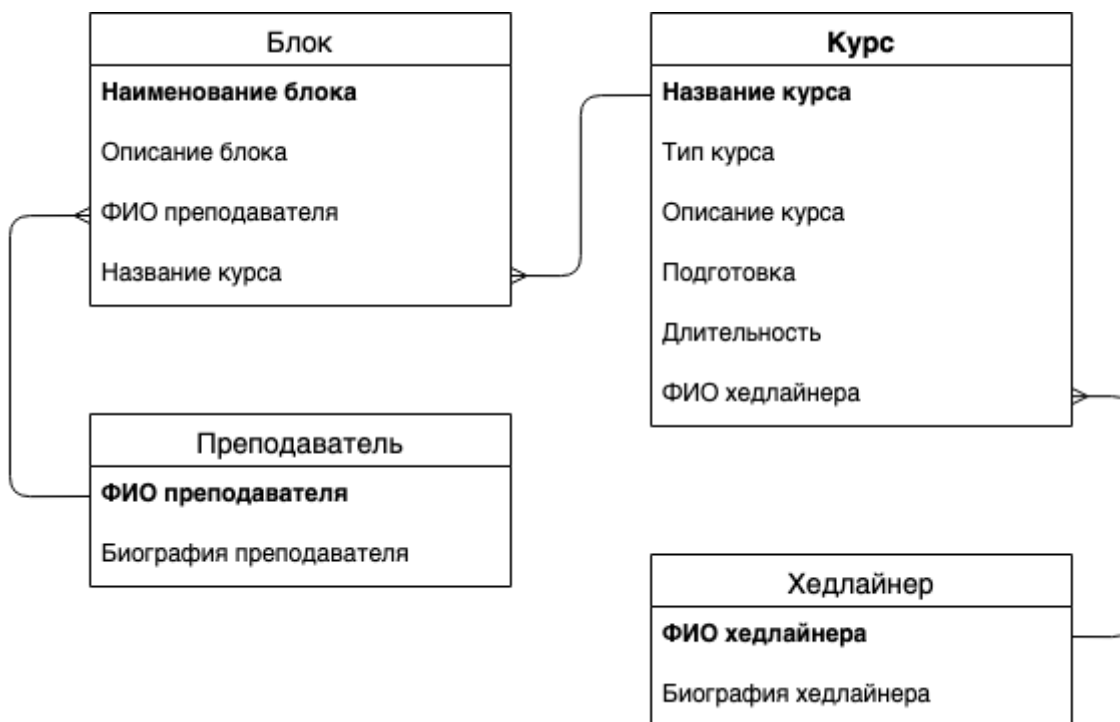


Таблица в 3НФ

Мы разделили таблицу на четыре новые сущности: **Блок**, **Курс**, **Преподаватель** и **Хедлайнер**, избавившись от транзитивных зависимостей. Связи выглядят следующим образом:

- У хедлайнера может быть много курсов
- В одном курсе может быть много блоков
- У одного преподавателя может быть много блоков

## > Вторая итерация нормализации таблицы

Теперь давайте посмотрим, что будет, если наша таблица с данными немного изменится, и как при этом нам придется пересмотреть нашу диаграмму.

Скачать её можно [здесь](#).

В таблице появилась новая строка. Это блок с именем *Визуализация* в курсе *Pre Data Engineer*. Однако блок с таким названием уже существует – он есть в курсе *Data Engineer*. Тем самым, теперь нельзя утверждать, что все значения в атрибуте **Наименование блока** – уникальны. Этот атрибут не может быть единственной частью ключа.

Чтобы гарантировать уникальность строки, мы должны использовать составной ключ – **{Название курса, Наименование блока}**.

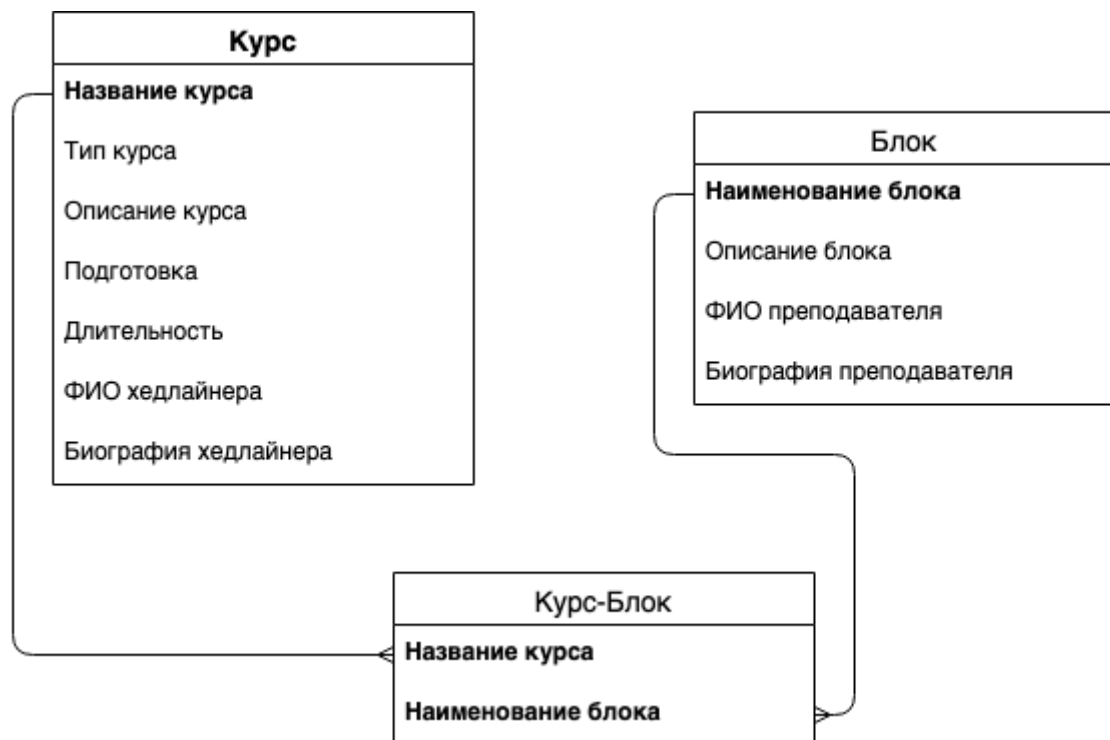
Table
<b>Название курса</b>
Тип курса
Описание курса
Подготовка
Длительность
ФИО хедлайнера
Биография хедлайнера
<b>Наименование блока</b>
Описание блока
ФИО преподавателя
Биография преподавателя

Исходная таблица с выбранным первичным ключом

---

## Несоответствие 2НФ

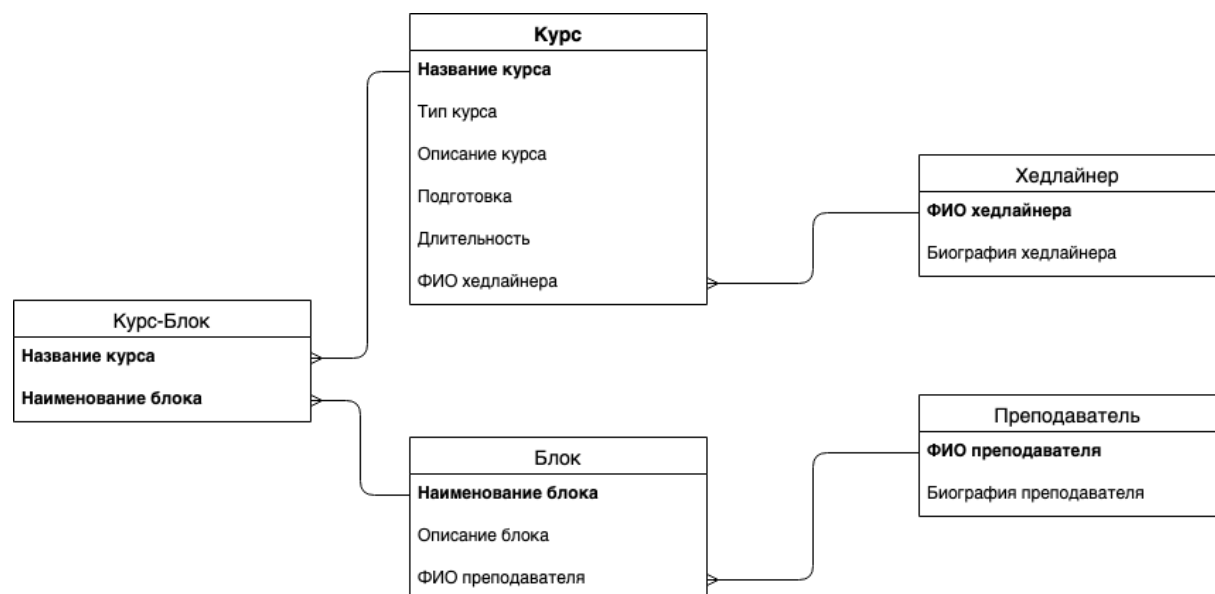
Так как название курса стало частью первичного ключа, то некоторые атрибуты теперь **зависят от части первичного ключа** – таблица не удовлетворяет 2НФ. Разделим таблицу для разрешения этой проблемы.



Таблица, приведенная во вторую нормальную форму

## Приведение к 3НФ

Для приведения схемы к 3НФ осталось выделить сущности **преподаватель** и **хедлайнер**, аналогично тому, как мы сделали это в предыдущем шаге:



Таблица, приведенная в третью нормальную форму

## > Третья итерация нормализации таблицы

Наша таблица снова немного поменялась – [вот тут](#).

Добавилась новая строка – теперь блок *ETL* ведет не один, а два преподавателя. Получается, что связка {Название курса, Наименование блока} не является уникальной и не может быть первичным ключом.

Дополним первичный ключ атрибутом **ФИО преподавателя**:

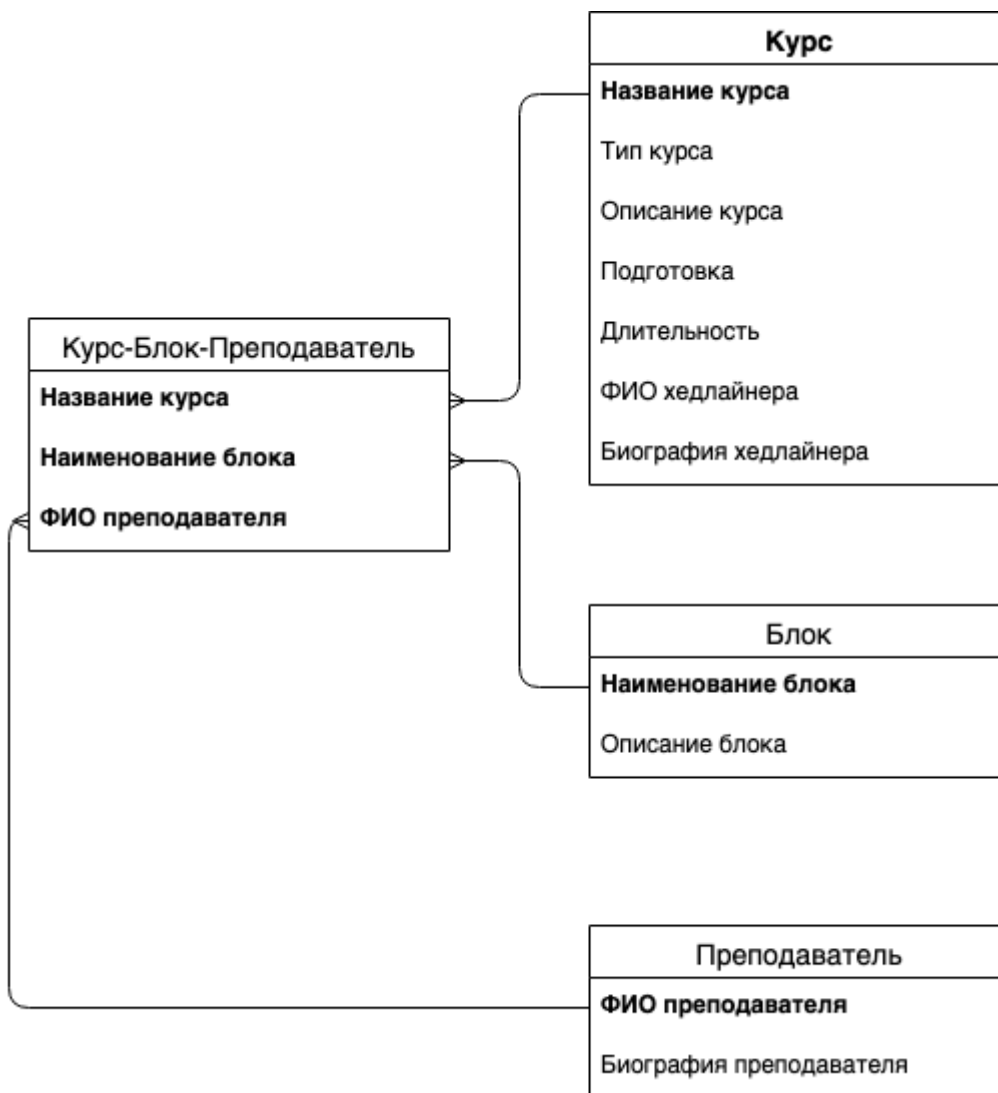
Table
<b>Название курса</b>
Тип курса
Описание курса
Подготовка
Длительность
ФИО хедлайнера
Биография хедлайнера
<b>Наименование блока</b>
Описание блока
<b>ФИО преподавателя</b>
Биография преподавателя

Таблица с выбранным первичным ключом

---

### Приведение к 2НФ

Очевидно, что наша таблица имеет такие же проблемы, как и в предыдущем шаге – некоторые атрибуты зависят только от части первичного ключа. Исправим это.

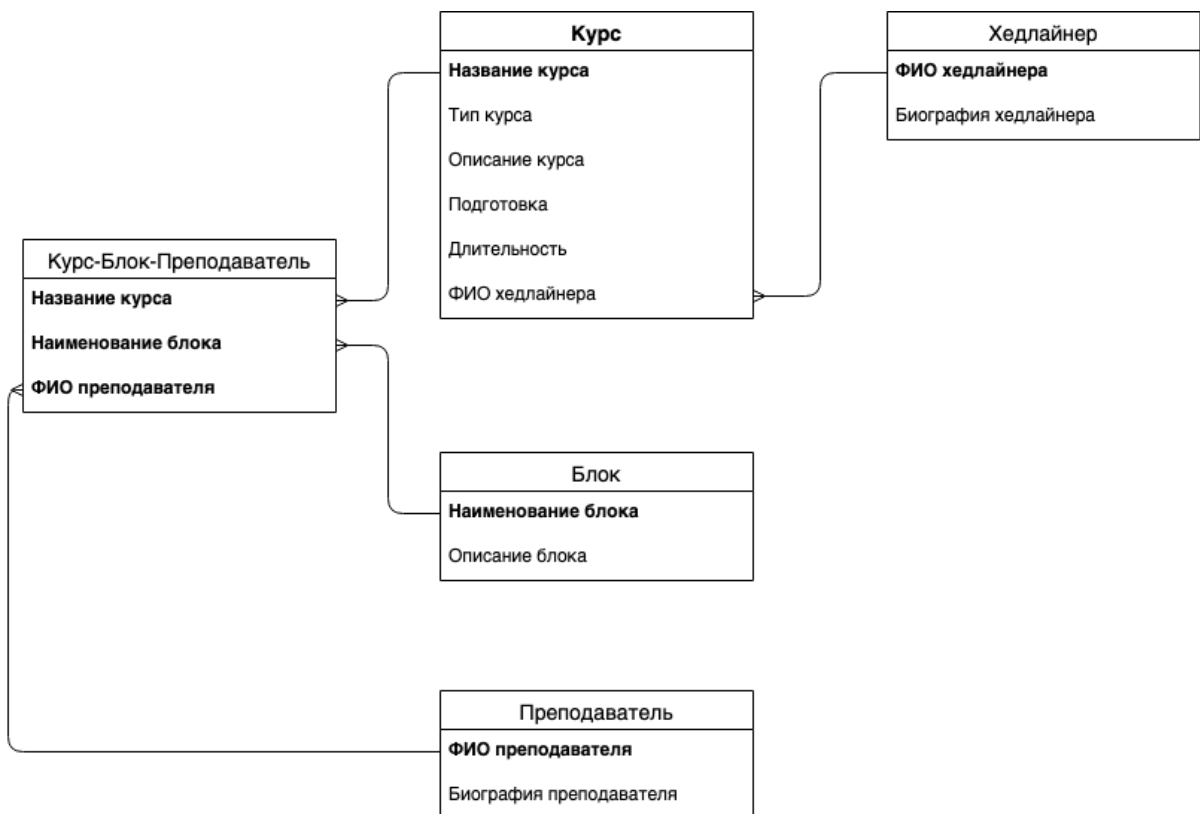


Таблица, приведенная ко второй нормальной форме

## Приведение к 3НФ

Для приведения схемы к третьей нормальной форме необходимо выделить отдельную сущность **Хедлайнер**.

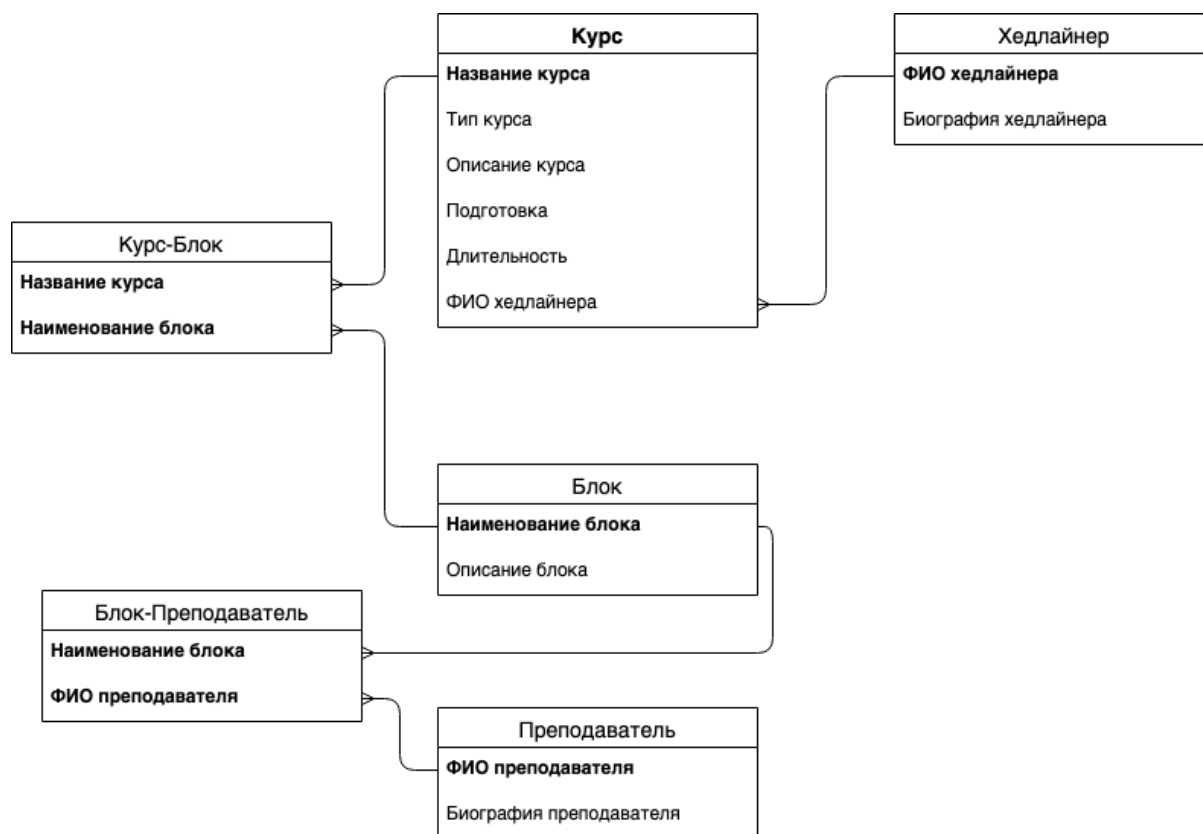




Таблица, приведенная к третьей нормальной форме

## Приведение к 4НФ

Для приведения схемы к четвертой нормальной форме необходимо разбить нашу связь **Курс-Блок-Преподаватель**, так как это троичная связь. Например, мы не можем добавить блок, если мы еще не нашли преподавателя. Разобьем эту связь на две связи многие-ко-многим: **Курс-Блок** и **Блок-Преподаватель**.



Таблица, приведенная к четвертой нормальной форме

## > Итоги

Важно понимать, что если смотреть только на какую-то часть данных, то велик риск неправильно спроектировать таблицы. Во время практики мы добавляли лишь по одной строке в исходную таблицу, однако это кардинально меняло наше представление о структуре данных, и ранее нормализованные таблицы теряли актуальность.

При проектировании важно смотреть не только на сами данные, но и на общее понимание бизнес-процессов (природу самих данных). Например, несмотря на то, что на имеющихся семплах данных мы не встретили такой случай, вполне вероятна ситуация, когда у одного курса не один, а несколько хедлайнеров. Мы не учли это в своих схемах.

Скачать файл со всеми построенными схемами можно [здесь](#), а открыть [здесь](#).