

Нормализация структуры БД: анализ по критериям из лекции “[1.3. САР-теорема и нормализация](#)”

При нормализации структуры БД к ЗНФ, я опиралась на критерии нормальных форм, которые были приведены в лекции «[1.3. САР-теорема и нормализация](#)» на LMS.

**Для удобства привожу их ниже:**

**1НФ:** Для того чтобы таблица была в 1НФ все атрибуты в таблице должны быть простыми, все сохраняемые данные на пересечении столбцов и строк — содержать лишь скалярные значения.

**2НФ:** Отношения будут соответствовать 2НФ, если сама БД находится в 1НФ, а каждый столбец, который не является ключом, зависит от первичного ключа.

**3НФ:** Таблица должна находиться в 2НФ, плюс любой столбец, который не является ключом, должен зависеть лишь от первичного ключа.

Исходная структура БД состоит из трех таблиц:

- customers
- transactions
- products

и доступна по [ссылке](#)

Ниже привожу проверку каждой таблицы на соответствие критериям из лекции, с обоснованием (если это необходимо) и внесением необходимых изменений для приведения к требуемой нормальной форме, начиная с 1НФ и заканчивая ЗНФ.

Таблица products

Атрибуты таблицы:

```
product_id int [pk]

brand varchar
product_line varchar
product_class varchar
product_size varchar
list_price decimal
standard_cost decimal
```

*Проверка на соответствие 1НФ*

Все атрибуты таблицы являются простыми, каждая ячейка содержит только одно значение.

Составных или повторяющихся групп нет, - следовательно, таблица products находится в 1НФ.

*Проверка на соответствие 2НФ*

В таблице products:

- brand
- product\_line
- product\_class
- product\_size
- list\_price
- standard\_cost

всё однозначно определяется товаром. Если известен product\_id, то легко можно определить каждый из оставшихся атрибутов таблицы. Таким образом, все поля зависят от ключа целиком и **таблица products находится во 2НФ**.

#### **Примечание**

В процессе анализа у меня возникла гипотеза, что между атрибутами:

- product\_line
- product\_class
- product\_size

может существовать функциональная зависимость.

Например, что конкретная линейка товара (product\_line) всегда определяет его класс (product\_class) или размер (product\_size).

Для проверки этой гипотезы я просмотрела реальный набор данных, и обнаружила следующее:

- одно и то же значение product\_line может встречаться с разными product\_class;
- одно и то же значение product\_class встречается с разными product\_size;
- комбинации значений повторяются, но не образуют строгой однозначности.

То есть по фактическим данным нельзя однозначно определить один атрибут по другому, и следовательно никакой функциональной зависимости между этими полями нет.

Это означает, что эти атрибуты не могут быть причиной нарушения 2НФ и 3НФ и их дополнительное выделение в отдельные таблицы не требуется.

#### **Проверка на соответствие 3НФ**

Следующим шагом проверяю, существует ли транзитивная зависимость между неключевыми полями.

Для этого необходимо определить, существует ли такая ситуация, при которой один неключевой атрибут определяет другой неключевой атрибут. Выше уже отмечалось, что предположения о возможной зависимости между product\_line, product\_class и product\_size не подтвердились.

Дополнительно хочу зафиксировать, что остальные атрибуты:

- brand,
- list\_price,
- standard\_cost

также не выступают детерминантами для других атрибутов, - следовательно, **таблица products находится в 3НФ**.

#### Таблица transactions

Атрибуты таблицы:

```
transaction_id int [pk]
```

```
product_id int
```

```
customer_id int  
transaction_date date  
online_order boolean  
order_status varchar
```

#### Проверка на соответствие 1НФ

Все атрибуты таблицы являются простыми, каждая ячейка содержит только одно значение.

Составных или повторяющихся групп нет, - следовательно, таблица transactions находится в 1НФ.

#### Проверка на соответствие 2НФ

В соответствии с определением из лекции, таблица находится в 2НФ, если все неключевые атрибуты полностью зависят от первичного ключа.

Первичный ключ — transaction\_id. Он простой, и, как и в случае с таблицей products, это означает, что частичная зависимость здесь в принципе невозможна.

Таким образом, по ключу transaction\_id можно определить все остальные атрибуты. Таблица transactions находится во 2НФ.

#### Проверка на соответствие 3НФ

Следующим шагом проверяю, существует ли транзитивная зависимость между неключевыми полями.

Проверяю возможные связи между атрибутами:

- product\_id не определяет order\_status
- customer\_id не определяет online\_order
- transaction\_date не определяется ни product\_id, ни customer\_id
- статус заказа не зависит от способа оформления (online\_order)
- и наоборот

Таким образом, ни один неключевой столбец не выступает детерминантом для другого неключевого столбца.

Все неключевые атрибуты логически независимы друг от друга и зависят только от первичного ключа — transaction\_id, - следовательно, таблица transactions находится в 3НФ.

Таблица customers

Атрибуты таблицы:

```
customer_id int [pk]  
  
first_name varchar  
last_name varchar  
gender varchar  
DOB date  
job_title varchar  
job_industry_category varchar  
wealth_segment varchar  
deceased_indicator varchar
```

```
owns_car varchar  
address varchar  
postcode varchar  
state varchar  
country varchar  
property int
```

#### *Проверка на соответствие 1НФ*

Все атрибуты таблицы *customers* являются простыми, а значения — скалярными.  
Вложенных структур нет, повторяющихся групп столбцов также нет, - следовательно,  
*таблица customers находится в 1НФ.*

#### *Проверка на соответствие 2НФ*

Первичный ключ — простой *customer\_id*, поэтому, как и в двух предыдущих таблицах,  
частичных зависимостей возникнуть не может.  
Каждый неключевой атрибут описывает конкретного клиента, а значит логически  
зависит от *customer\_id*, - следовательно, *таблица customers находится во 2НФ*

#### *Проверка на соответствие 3НФ*

На этом шаге становится заметным нарушение. В таблице присутствует транзитивная  
зависимость, что противоречит определению 3НФ.

Столбец *postcode* однозначно определяет:

- *state*
- *country*

На практике один и тот же почтовый индекс всегда принадлежит одному штату и одной  
стране.

Это означает, что существует зависимость:

*postcode* → *state*  
*postcode* → *country*

Отсюда следует транзитивная зависимость:

*customer\_id* → *postcode* → *state, country*

что противоречит критериям 3НФ: неключевые поля *state* и *country* зависят не от  
первичного ключа, а от другого неключевого поля.

Следовательно, *таблица customers в текущем виде не находится в 3НФ.*

#### *Корректировка таблицы *customers* для соответствия 3НФ*

Для устранения транзитивной зависимости таблица была переработана следующим  
образом:

1. Создана отдельная таблица *postcodes*:

```
2. Table postcodes {  
3.   postcode varchar [pk]  
4.   state varchar  
5.   country varchar  
6. }
```

2. Из таблицы *customers* удалены столбцы *state* и *country*, оставлен только  
*postcode*;
3. Добавлена связь *customers.postcode* > *postcodes.postcode*.

После вынесения зависимых атрибутов в отдельную таблицу транзитивная зависимость устранена, - следовательно, таблица `customers` стала находится в ЗНФ.

**Обновленная структура БД, нормализованная до ЗНФ, доступна по [ссылке](#).**