**Приведение к Первой нормальной форме (1НФ)**

**Требования 1НФ:**

1. Все столбцы должны содержать атомарные (неделимые) значения.
2. У таблицы должен быть **первичный ключ**.
3. Все записи должны быть уникальными.

**Проблемы в исходных таблицах:**

* Поля brand, product\_line, product\_class, product\_size в transactions **могут содержать повторяющиеся данные**.
* В customers **не хватает атомарности** (например, address объединяет улицу, город, штат и индекс в один столбец).
* Возможно, **один клиент делает несколько покупок**, что потенциально приводит к дублированию данных о клиенте в transactions.

**Исправления:**

* **Разделение address на несколько полей** (street, city, state, postcode).
* **Гарантируем, что каждая строка в transactions и customers уникальна**.
* **Поля brand, product\_line, product\_class выносим в отдельную таблицу products**.

**Приведение ко Второй нормальной форме (2НФ)**

**Требования 2НФ:**

1. База данных должна быть в **1НФ**.
2. **Все неключевые атрибуты** должны **зависеть только от первичного ключа**.

**Проблемы в 1НФ:**

* В transactions есть **избыточность**: brand, product\_line, product\_class, product\_size зависят **не от транзакции, а от продукта**.
* Поля job\_title, job\_industry\_category, wealth\_segment в customers можно вынести в отдельную таблицу, так как они относятся к категории, а не уникальному клиенту.

**Исправления:**

* **Создаем таблицу products и переносим туда информацию о товарах**.
* **Создаем таблицу job\_categories и связываем её с customers**.

**Шаг 3: Приведение к Третьей нормальной форме (3НФ)**

**Требования 3НФ:**

1. База данных должна быть в **2НФ**.
2. **Все неключевые атрибуты не должны зависеть друг от друга**, а только от **первичного ключа**.

**Проблемы в 2НФ:**

* В customers поле state зависит от postcode, а country зависит от state. То есть **некоторые атрибуты зависят друг от друга**.
* В transactions поля order\_status, online\_order можно вынести в таблицу **order\_statuses**.

**Исправления:**

* **Выносим state и country в отдельную таблицу locations**.
* **Создаем таблицу order\_statuses**, чтобы уменьшить дублирование данных.

После этого база данных соответствует **3НФ**.

**Финальная структура (в 3НФ)**

1. **customers (Клиенты)**
   * customer\_id (PK)
   * first\_name
   * last\_name
   * gender
   * DOB
   * job\_id (FK → job\_categories.job\_id)
   * wealth\_segment
   * deceased\_indicator
   * owns\_car
   * location\_id (FK → locations.location\_id)
   * property\_valuation
2. **job\_categories (Категории профессий)**
   * job\_id (PK)
   * job\_title
   * job\_industry\_category
3. **locations (Локации)**
   * location\_id (PK)
   * address
   * postcode
   * state
   * country
4. **transactions (Транзакции)**
   * transaction\_id (PK)
   * product\_id (FK → products.product\_id)
   * customer\_id (FK → customers.customer\_id)
   * transaction\_date
   * online\_order
   * order\_status\_id (FK → order\_statuses.order\_status\_id)
   * list\_price
   * standard\_cost
5. **order\_statuses (Статусы заказов)**
   * order\_status\_id (PK)
   * order\_status
6. **products (Продукты)**
   * product\_id (PK)
   * brand
   * product\_line
   * product\_class
   * product\_size