О нормальных формах исходных таблиц:

**1. Первая нормальная форма (1НФ):**

В исходных таблицах данные уже находятся в 1НФ , тк все атрибуты уже атомарны (не содержат повторяющихся групп или составных значений).

**2. Вторая нормальная форма (2НФ):**

Мы уже знаем, что таблицы находятся в 1НФ. Тогда можно исследовать на соответствие НФ2, то есть цель чтобы все неключевые атрибуты полностью завесили от первичного ключа:

В таблице транзакций (transaction):

Первичный ключ: transaction\_id. Однако атрибуты product\_id, customer\_id, brand, product\_line, product\_class, product\_size, list\_price, standard\_cost зависят от product\_id, а не от transaction\_id. что нарушает 2НФ.

В таблице клиентов (customer):

Первичный ключ: customer\_id. Все атрибуты зависят от customer\_id, поэтому таблица уже находится в 2НФ.

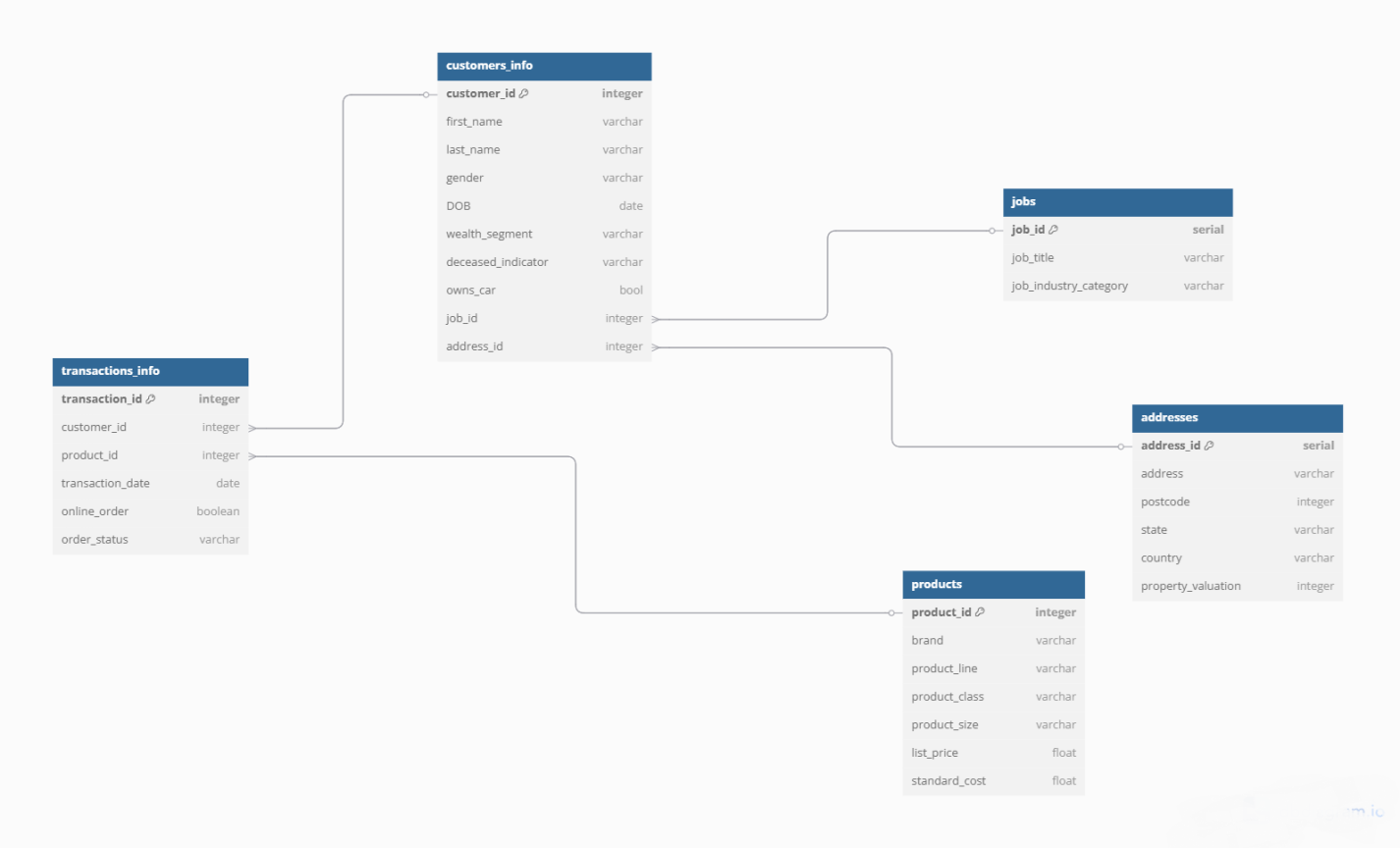
Как итог: таблица транзакций (transaction) находится в 1НФ и не может в текущем виде быть приведена к 3НФ, тк для этого нужно сначала привести ее к 2НФ. Таблица customer уже во второй нормальной форме, можно проверить на третью.

**3. Третья нормальная форма (3НФ):**

В таблице клиентов:

Атрибуты job\_title, job\_industry\_category могут зависеть от job\_id, а не напрямую от customer\_id. Это нарушает 3НФ.

Сделаем нормализованную структуру базы данных:



Для образования нормализованной структуры были добавлены новые атрибуты: job\_id и address\_id для таблиц jobs и address. Они уникальные идентификаторы, которые автоматически генерируются с помощью типа SERIAL, и в каждой из этих таблиц есть ограничения на уникальность: уникальные сочетания job\_title, job\_industry\_category для jobs и address, postcode, state, country, property\_valuation для addresses

Нормализованная структура базы данных:

1. Таблица транзакций (transactions\_info):

* transaction\_id (PK)
* customer\_id (FK)
* product\_id (FK)
* transaction\_date
* online\_order
* order\_status

1. Таблица клиентов (customers\_info):

* customer\_id (PK)
* first\_name
* last\_name
* gender
* DOB
* job\_id (FK)
* wealth\_segment
* deceased\_indicator
* owns\_car
* address\_id (FK)

1. Таблица продуктов (products):

* product\_id (PK)
* brand
* product\_line
* product\_class
* product\_size
* list\_price
* standard\_cost

1. Таблица работы (jobs):

* job\_id (PK)
* job\_title
* job\_industry\_category

1. Таблица адресов (addresses):

* address\_id (PK)
* address
* postcode
* state
* country
* property\_valuation

База данных нормализована до 3НФ: устранены избыточность и транзитивные зависимости.

Обоснование:

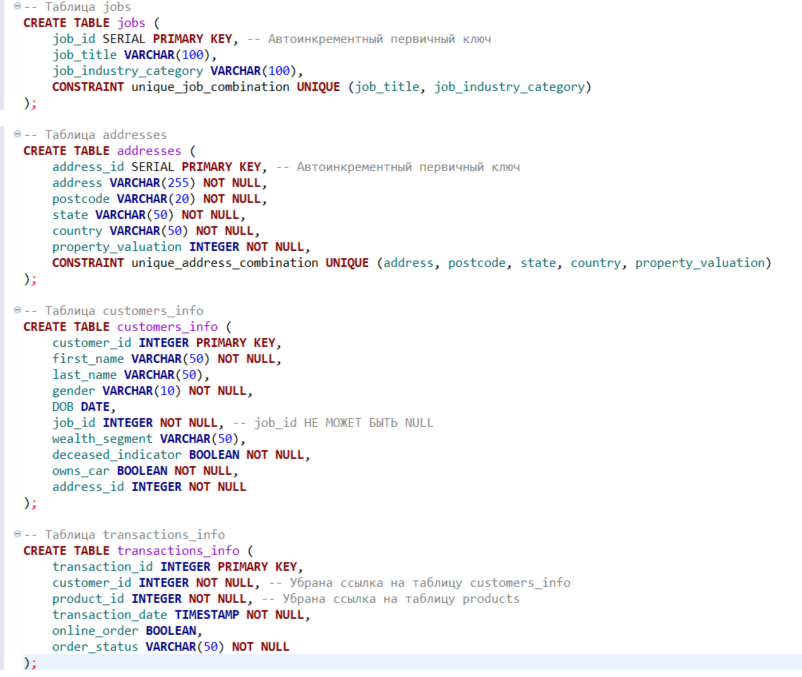
transactions\_info: Вынесли информацию о продуктах в отдельную таблицу, чтобы устранить зависимость от product\_id.

customers\_info: Вынесли информацию о работе и адресе в отдельные таблицы, чтобы устранить транзитивные зависимости.

products: Создали отдельную таблицу для хранения информации о продуктах.

jobs: Создали отдельную таблицу для хранения информации о работе.

addresses: Создали отдельную таблицу для хранения информации об адресах



Загрузим не все данные, а маленькую выборку для быстроты и удобства, с помощью просто вставки нескольких строк

Таблица transactions\_info:

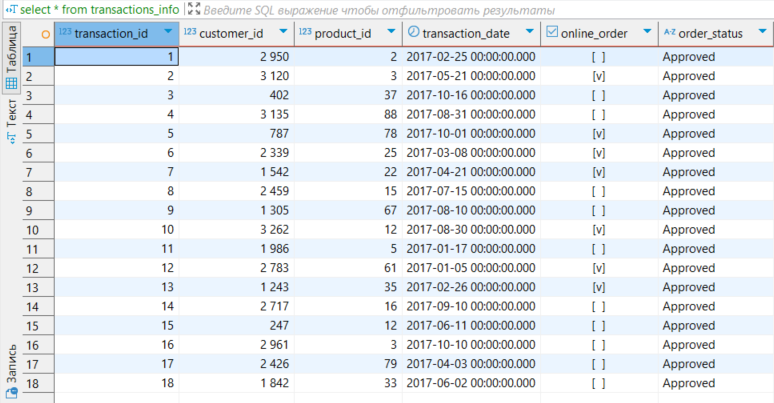


Таблица customers\_info:

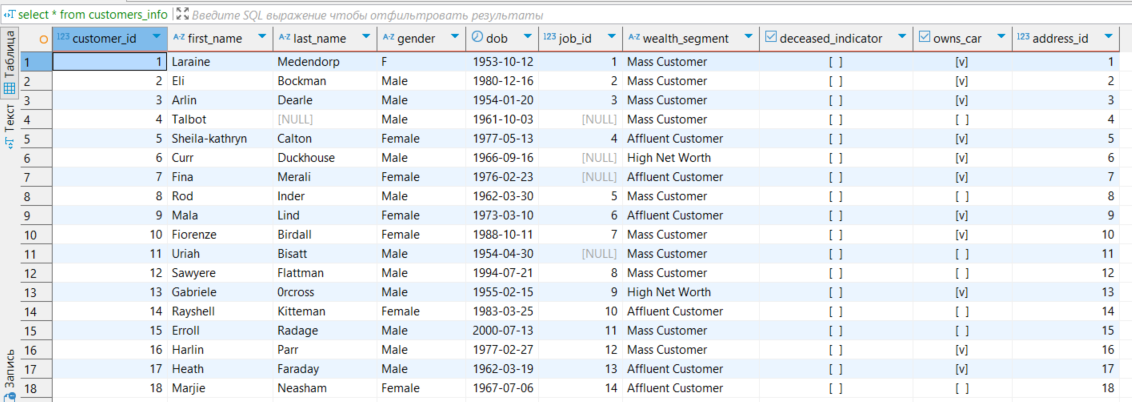


Таблица jobs:

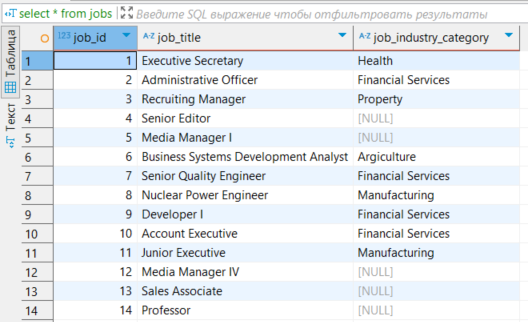


Таблица addresses:

