Домашнее задание N°1 "Создание и нормализация базы данных"
по предмету "Системы хранения обработки данных"
студента Ильиных А.А.

Формулировка задания
Дан файл с данными по клиентам и транзакциям: customer_and_transaction.xlsx.

Необходимо выполнить следующие пункты:
(2 балла) Продумать структуру базы данных и отрисовать в редакторе.
(2 балла) Нормализовать базу данных (1НФ — 3НФ), описав, к какой нормальной форме приводится таблица и почему таблица в этой нормальной форме изначально не находилась.
(3 балла) Создать все таблицы в DBeaver, указав первичные ключи к таблицам, правильные типы данных, могут ли поля быть пустыми или нет (использовать команду CREATE TABLE).
(3 балла) Загрузить данные в таблицы в соответствии с созданной структурой (использовать команду INSERT INTO или загрузить файлы, используя возможности инструмента DBeaver; в случае загрузки файлами приложить скрины, что данные действительно были залиты).

1. Задание
Выполнено, отрисованно в https://dbdiagram.io/d.
Из имеющего файла с двумя листами, дополнительна выделенна таблицы product, job с с ключами id_product_big, job_id (признаки созданы вновь, суррогатные ключи) для разделения статуса покупки, и самого продукта и выделения словаря в отдельную таблицу по типу работы и должности.

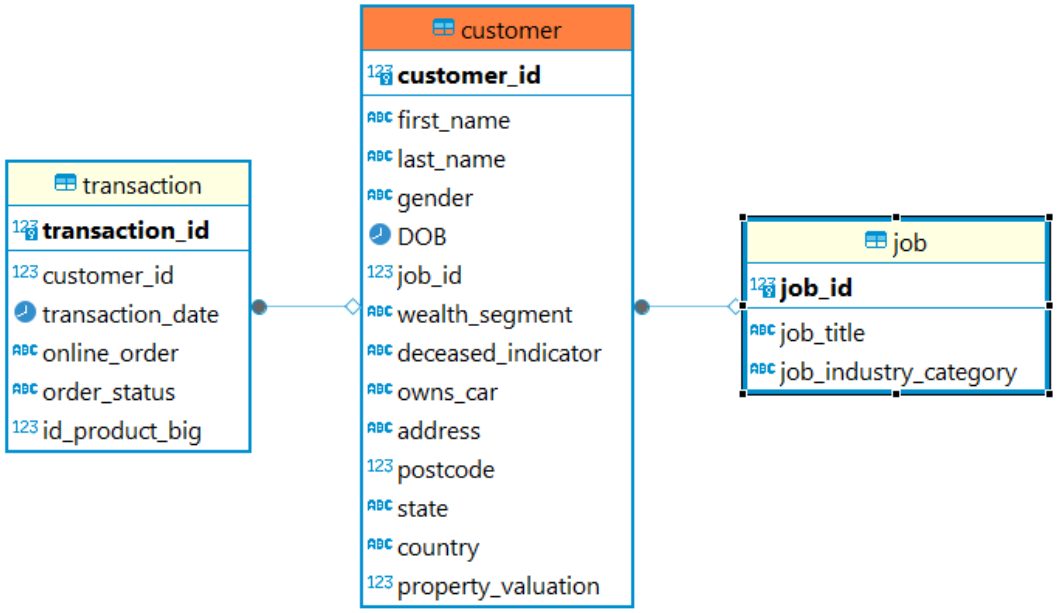
```
Table customer {
customer_id integer [primary key]
first_name varchar
last_name varchar
gender varchar
DOB timestamp
job_id integer
wealth_segment varchar
deceased_indicator varchar
owns_car varchar
address varchar
postcode integer
state varchar
country varchar
property_valuation integer
}

Table transaction {
transaction_id integer [primary key]
customer_id integer
transaction_date timestamp
online_order varchar
order_status varchar
```

id_product_big integer
}

Table product {
id_product_big integer [primary key]
product_id integer
brand varchar
product_line varchar
product_class varchar
product_size varchar
standard_cost float
list_price float
}

Table job {
job_id integer [primary key]
job_title varchar
job_industry_category varchar
}

Ref: transaction.customer_id > customer.customer_id
Ref: transaction.id_product_big > product.id_product_big
Ref: customer.job_id > job.job_id



2. Задание

Первая нормальная форма
Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.
В базе данных не должно быть дубликатов и составных данных.

Первоначально таблица находится в Первой нормальной форме, так как каждая строка в таблицах transaction, customer уникальна и не содержит составных данных.

Вторая нормальная форма
Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК).
Если упростить: у каждой записи в базе данных должен быть первичный ключ. Первичный ключ — это элемент записи, который не повторяется в других записях.

Также база находится во Второй нормальной форме, в таблице transaction, customer имеются столбцы с уникальными ключами transaction_id и customer_id соотвественно.

Третья нормальная форма
Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. В записи не должно быть столбцов с неключевыми значениями, которые зависят от других неключевых значений.

При анализе данных в исходном файле в Excel:

- в таблице *transaction* имеются зависимые столбцы в комбинации следующих столбцов: product_id, product_line, product_class, product_size, brand, standard_cost, list_price.

- в таблице *customer* имеются зависимые столбцы в комбинации следующих столбцов: job_title job_industry_category. job_id

Для приведения к Третьей нормальной форме:

- вынесем данные столбцы в новую таблицу product. И свяжем новым ключем id_product_big (создаем его из хеша переносимых столбцов, построчно).

- вынесем данные столбцы в новую таблицу job. И свяжем новым ключем job_id (создаем его из хеша переносимых столбцов, построчно).

После вынесения данных таблиц база данных будет преведена к третьей нормальной форме, так не будет столбцов которые зависят напрямую от других значений.

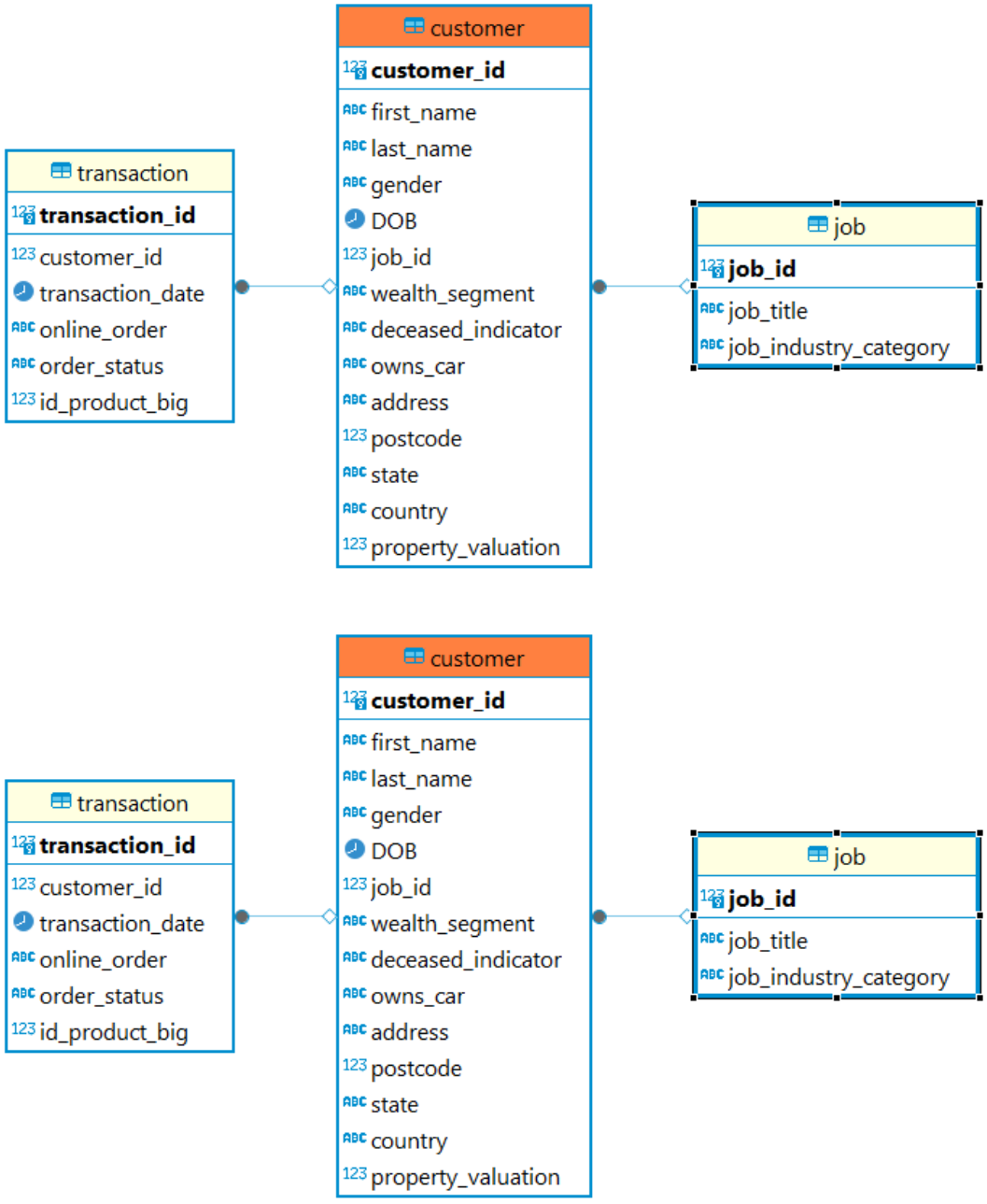
3. Задание.
Создаем базу запросом SQL
Создать все таблицы в DBeaver, указав первичные ключи к таблицам, правильные типы данных, могут ли поля быть пустыми или нет (использовать команду CREATE TABLE).

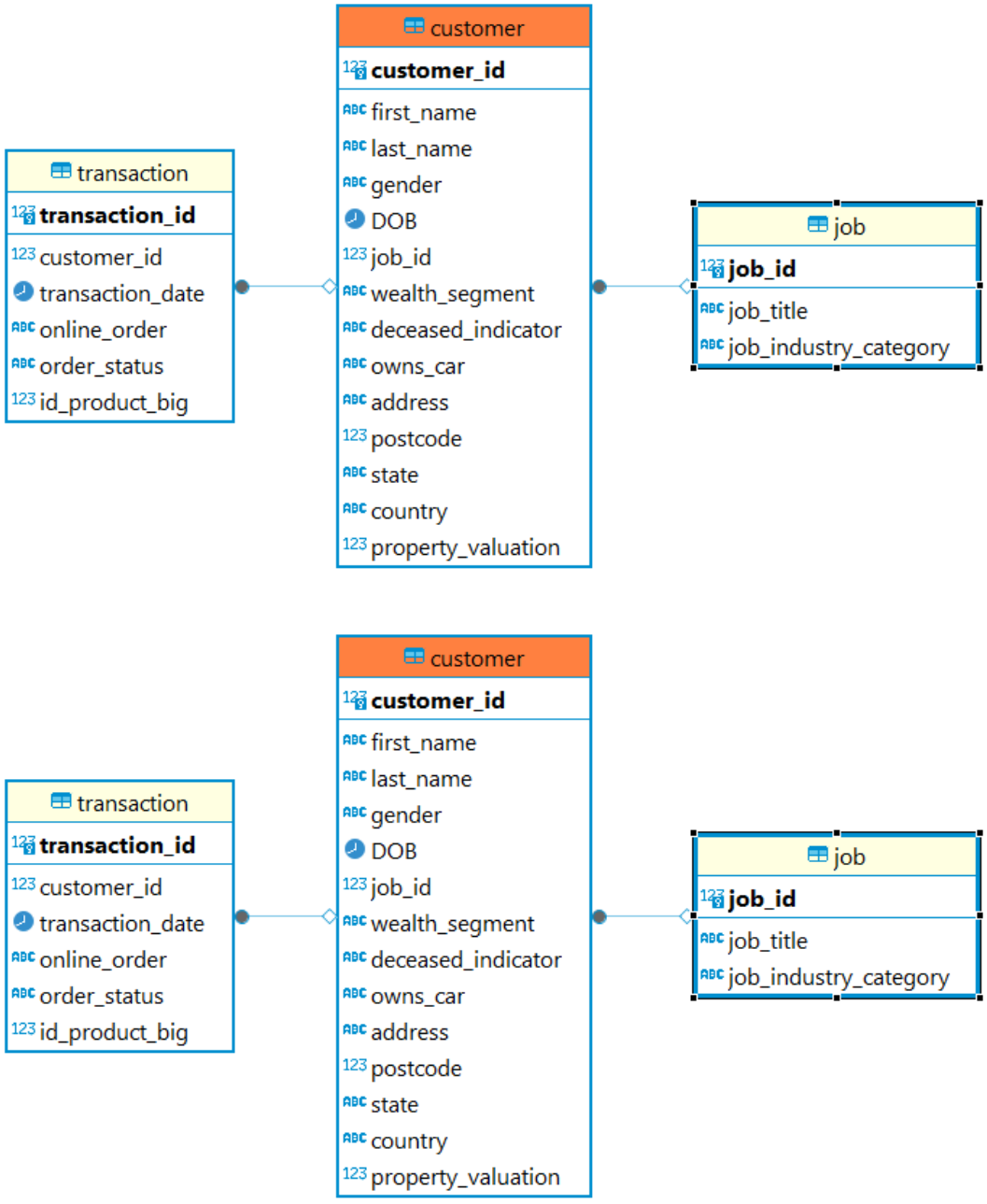
```
CREATE TABLE job (
  "job_id" integer PRIMARY KEY
  ,"job_title" varchar
  ,"job_industry_category" varchar
  ,UNIQUE("job_id")
);

CREATE TABLE product (
  "id_product_big" integer PRIMARY KEY
```

```
  ,"product_id" integer
  ,"brand" varchar
  ,"product_line" varchar
  ,"product_class" varchar
  ,"product_size" varchar
  ,"standard_cost" float
  ,"list_price" float
  ,UNIQUE("id_product_big")
);

CREATE TABLE customer (
  "customer_id" integer PRIMARY KEY
  ,"first_name" varchar
  ,"last_name" varchar
  ,"gender" varchar
  ,"DOB" timestamp
  ,"job_id" integer
  ,"wealth_segment" varchar
  ,"deceased_indicator" varchar
  ,"owns_car" varchar
  ,"address" varchar
  ,"postcode" integer
  ,"state" varchar
  ,"country" varchar
  ,"property_valuation" integer
  ,UNIQUE("customer_id")
  ,FOREIGN KEY ("job_id") REFERENCES job("job_id")
);

CREATE TABLE transaction (
  "transaction_id" integer PRIMARY KEY
  ,"customer_id" integer
  ,"transaction_date" timestamp
  ,"online_order" varchar
  ,"order_status" varchar
  ,"id_product_big" integer
  ,UNIQUE("transaction_id")
  ,FOREIGN KEY ("customer_id") REFERENCES customer("customer_id")
  ,FOREIGN KEY ("id_product_big") REFERENCES product("id_product_big")

);
```

CREATE TABLE job (
"job_id" integer PRIMARY KEY
,"job_title" varchar
,"job_industry_category" varchar
,UNIQUE("job_id")
);

```
CREATE TABLE product (
"id_product_big" integer PRIMARY KEY
,"product_id" integer
,"brand" varchar
,"product_line" varchar
,"product_class" varchar
,"product_size" varchar
,"standard_cost" float
,"list_price" float
,UNIQUE("id_product_big")
);

CREATE TABLE customer (
"customer_id" integer PRIMARY KEY
,"first_name" varchar
,"last_name" varchar
,"gender" varchar
,"DOB" timestamp
,"job_id" integer
,"wealth_segment" varchar
,"deceased_indicator" varchar
,"owns_car" varchar
,"address" varchar
,"postcode" integer
,"state" varchar
,"country" varchar
,"property_valuation" integer
,UNIQUE("customer_id")
,FOREIGN KEY ("job_id") REFERENCES job("job_id")
);

CREATE TABLE transaction (
"transaction_id" integer PRIMARY KEY
,"customer_id" integer
,"transaction_date" timestamp
,"online_order" varchar
,"order_status" varchar
,"id_product_big" integer
,UNIQUE("transaction_id")
,FOREIGN KEY ("customer_id") REFERENCES customer("customer_id")
,FOREIGN KEY ("id_product_big") REFERENCES product("id_product_big")
);
```


customer
customer_id
first_name
last_name
gender
DOB
job_id
wealth_segment
deceased_indicator
owns_car
address
postcode
state
country
property_valuation
transaction
transaction_id
customer_id
transaction_date
online_order
order_status
id_product_big
job
job_id
job_title
job_industry_category
customer
customer_id
first_name
last_name
gender
DOB
job_id
wealth_segment
deceased_indicator
owns_car
address
postcode
state
country
property_valuation
transaction
transaction_id
customer_id
transaction_date
online_order
order_status
id_product_big
job
job_id
job_title
job_industry_category



Подготовка данных:

1. Создал id_product_big с помощью сложения текста исходных столбцов (в экселе с помощью &) product_id, product_line, product_class, product_size, brand, standard_cost, list_price получилось примитивных хеш на 399 уникальных значений.
2. Создал копию листа, удалил дублирующие строки для создания словаря (хеш - id).

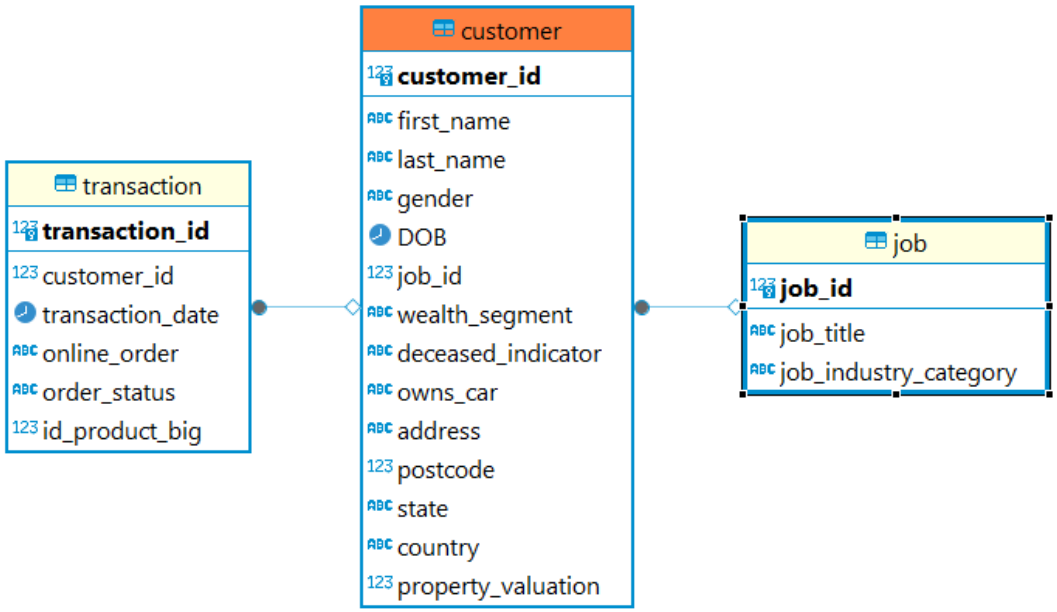
3. Для превращения хеша в число воспользовался функцией =ВПР(C1; A:B; 2; ЛОЖЬ). Здесь: C1 - это ячейка, значение которой вы хотите найти в столбце A. A:B - диапазон ячеек вашего словаря. 2 - это номер столбца, в котором находится значение, которое нужно вернуть (в данном случае, столбец B). ЛОЖЬ - означает точное совпадение.

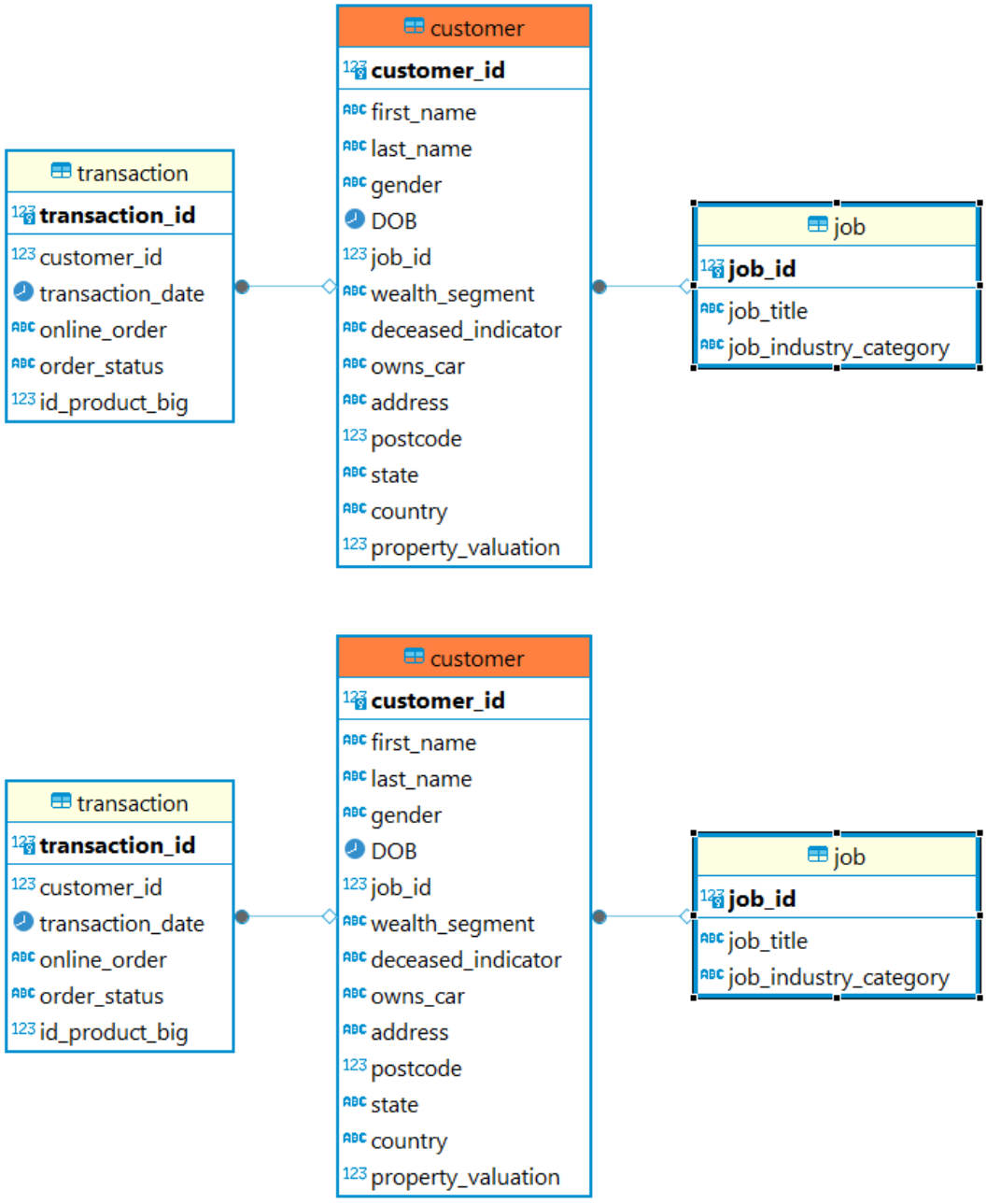
4. Для замены хеша на id воспользовался Формула для замены значений в столбце A по словарю из столбца D: "=ПОИСК(A1; D:E; 2; ЛОЖЬ)".

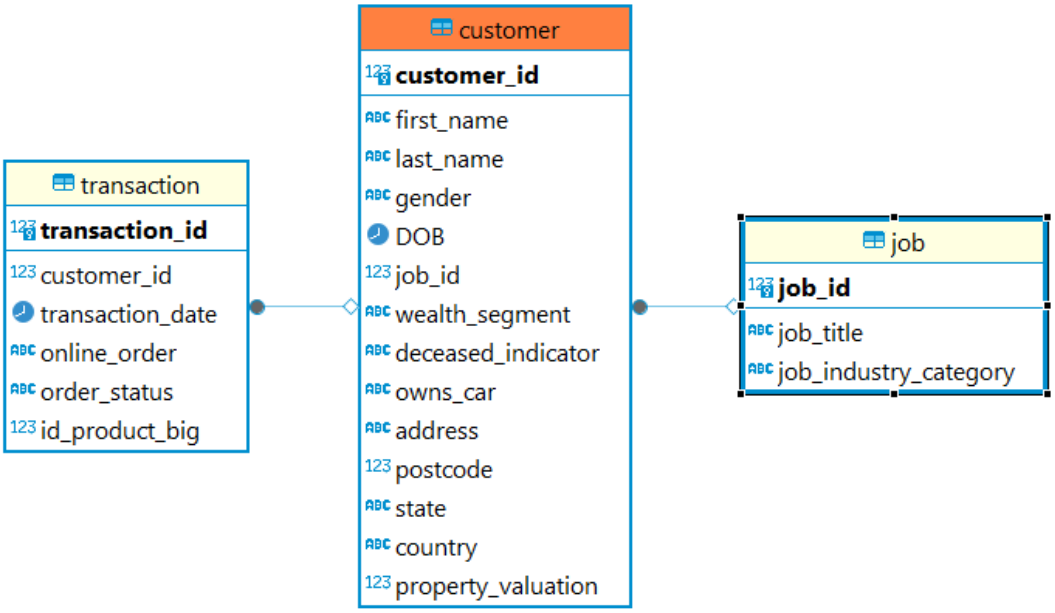
- Параметр A1 - это значение, которое мы ищем.
- Параметр D:E - это диапазон, в котором мы ищем значение (в данном случае, весь столбец D).
- Параметр 2 - это номер столбца в диапазоне D:D, который содержит значение, которое нужно вернуть. В данном случае, мы выбрали второй столбец (потому что у нас словарь замены с двумя столбцами: старые значения и новые значения).
- Параметр ЛОЖЬ - это параметр, который указывает функции ПОИСК использовать точное совпадение.

1. Аналогично по job_id.

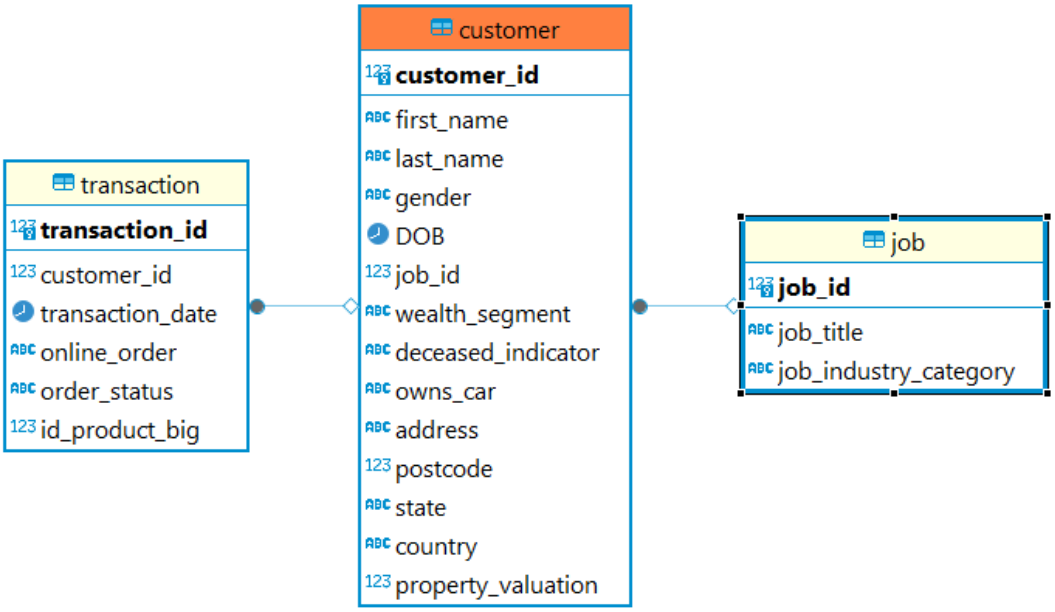
4. Задание.
Данные были залиты используя возможности инструмента DBeaver из cvs файлов ***(приложены в папке data)*** .
Возникли ошибки при загрузке таблицы trasaction. Имелись сломанное значение ключа customer_id = "5304".
Так как данная колонка ссылалась на таблицу customer, где customer_id заканчивался значением "4000".
Решил заменить customer_id = "5304" на "3304".

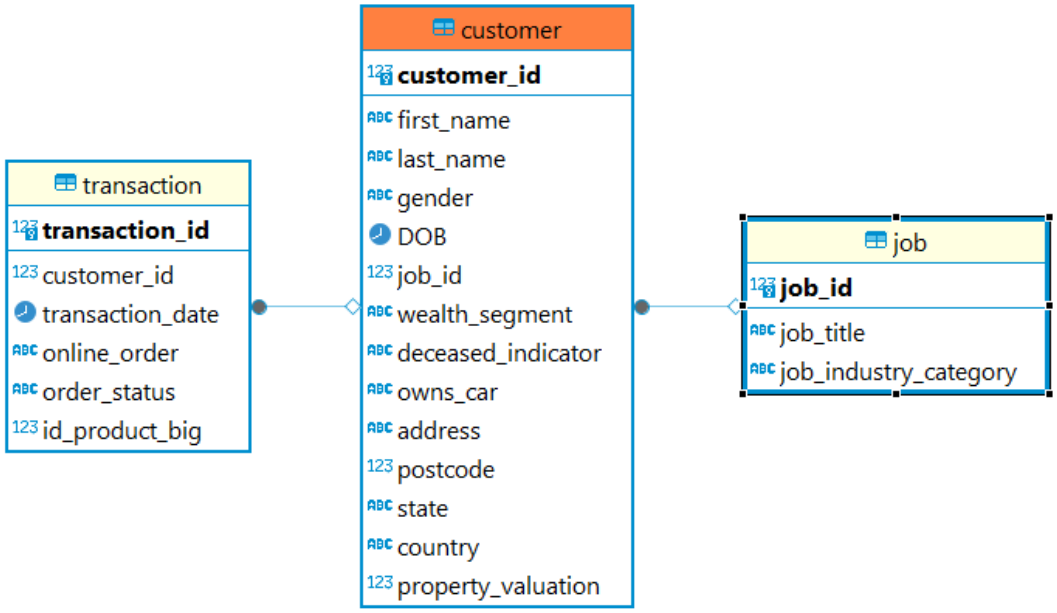



customer
customer_id
first_name
last_name
gender
DOB
job_id
wealth_segment
deceased_indicator
owns_car
address
postcode
state
country
property_valuation
transaction
transaction_id
customer_id
transaction_date
online_order
order_status
id_product_big
job
job_id
job_title
job_industry_category
customer
customer_id
first_name
last_name
gender
DOB
job_id
wealth_segment
deceased_indicator
owns_car
address
postcode
state
country
property_valuation
transaction
transaction_id
customer_id
transaction_date
online_order
order_status
id_product_big
job
job_id
job_title
job_industry_category



Дополнительно показал связи между таблицами.




customer
customer_id
first_name
last_name
gender
DOB
job_id
wealth_segment
deceased_indicator
owns_car
address
postcode
state
country
property_valuation
transaction
transaction_id
customer_id
transaction_date
online_order
order_status
id_product_big
job
job_id
job_title
job_industry_category