I Approached the assignment by working on the tables in the following order

1. Customer table
2. Vendor
3. Article category
4. articles
5. Order details
6. order details extended
7. Client\_address

For sake of organizing and to avoid redundancy, I created a separate table for the order details and order details extended as one customer can have many order IDs.

While I figured that the relation between customer ID and address is one too many I felt the reverse won't be possible as I felt we must have one to one relationship, I was not quite sure how to establish this. It works fine, but I am not fully convinced.

Few data types are handpicked for special reasons and I have mentioned them below

* **Customer Table**

1. I used the active field to know if the customer is still active and not using the account that is disabled
2. I Used the Enum data type to choose between two values (depending on the use case ) between 0 and 1
3. Chose phone number as BIGINT to store big integer

* **Vendor Table**

1. used is\_active to know if they are disabled or not
2. and a code for the vendor as well, like a user name.

* **Articles**

1. I added an option to use currency as I have seen it on many websites and I find it very useful. But I have declared Euro as the default currency in the article table.
2. Is\_visble: I added this is to give the vendor the flexibility to make it visible or not to the customers.

* **Article category**

1. To avoid redundancy I separated articles and article category table. As there can be many articles in one category. I tried implement grouping

* **Order :**

1. It has foreign key connections with several tables like customer,vendor,client\_address tables
2. it has foreign keys of the primary keys in the tables mentioned above like vendor\_id , customer\_id which are primary keys in vendor table and customer table respectively.

* **Client\_address**

1. I used the “default” as I wanted it to store a default value even in case of no input

* **Relationship**

Regarding the ER diagram, we can see the following relationship

1. One to many relationships

Implemented from customer table to client\_address.

Article\_category has many articles

Order details to order details extended

1. Many to one

Article to article\_category

Client address to the customer

1. Many to many

Order\_details to customer ,vendor ,articles and client\_address

I used the Faker library in python to dump data into the database. as I tried many times, the auto-increment value was not updated back to 0, to fix I had to the following command

DELETE FROM client\_address WHERE id\_cl\_address>0;

ALTER TABLE client\_address AUTO\_INCREMENT = 0;

SELECT \* FROM client\_address;

I still faced the same problem and found that I used the wrong command and replaced it with

ALTER TABLE `myStore`.`client\_address`

AUTO\_INCREMENT = 0 ;

I wanted to implement coupons table , due to time constraints I decided to drop the idea.