Practise - Big Data Analytics and Reasoning

The *Big Data Analytics Company* wants to analyze its e-commerce sails. Data are stored into their cluster and different formats. Here the file-system tree of their hdfs.

```
root
 ├── user
 └── e-commerce
          ├── customer_data
          │        ├── 01-customers.csv
          │        └── 02-customers.csv
          ├── seller_data
          │        └── sellers.csv
          ├── order_data
          │        ├── order_meta
          │        │        └── orders.csv
          │        └── items
          │                 └── order_items.csv
          ├── product_data
          │        └── products.csv
          └── category_translate.csv
```

Problem 1

Replicate e-commerce file system tree on your hdfs and define hive tables on top of this data.

Problem 2

Compute the following queries:

- For each customer, find the number of order with at least 2 items
- Find active customers for each year. A customer is active if it has at least three order in a given year.
- For each year and for each customer city, compute the total income for the company (i.e. the sum of the total price of each order)

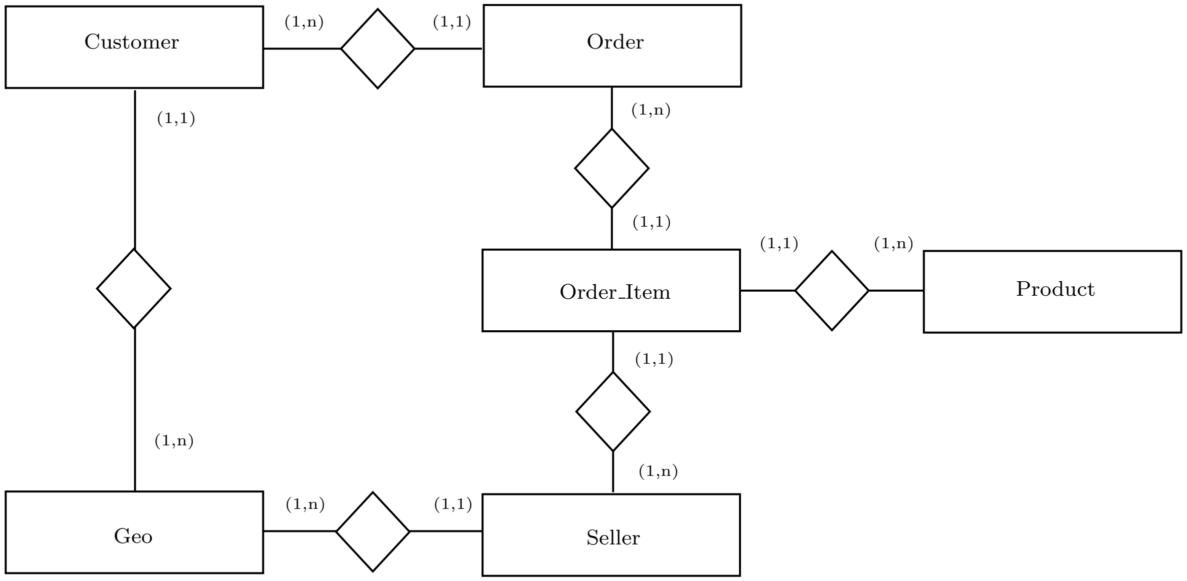
- Find the three most frequent categories (possibly english) among e-commerce product
- Find for each product the number of sold items and the total income
- Find product category (possibly english) compute of sold items and the total income

Problem 3

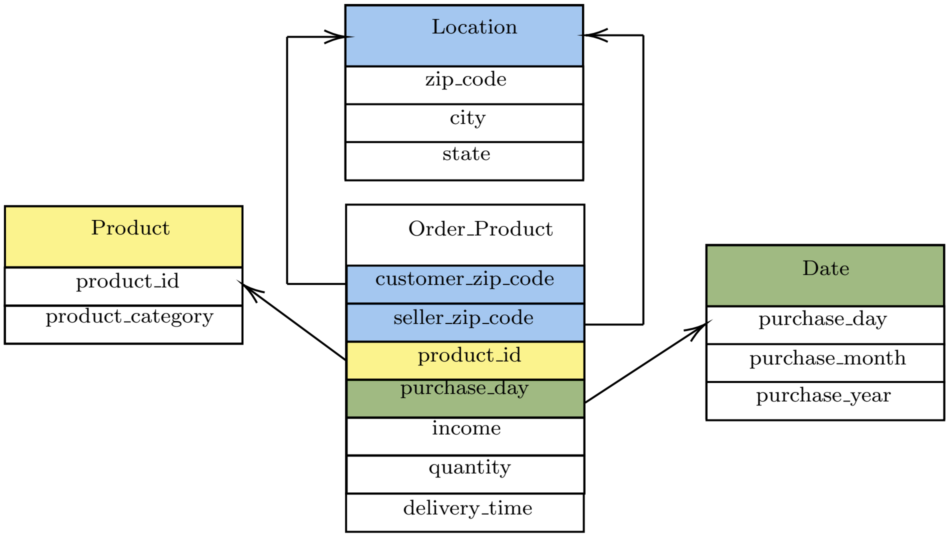
Design a simple java application that store incoming orders into the data source. The application should be able to read customers and products in a lazy fashion during orders creation. Note that no GUI is requested, just read and write on terminal.

Problem 4

The big data company wants to build a data warehouse for analyzing product sells. To this end, we can assume that data stored in the hdfs follow this schema the we refer to as **data source**:



Starting from the data source build the following fact schema, that we refer to as **e-commerce warehouse**:



The e-commerce warehouse should be populated from the data source by filtering out:

- All those products that have no assigned category
- All those orders having at least one product without an assigned category
- All those orders for which `delivered_carrier_date` or `delivered_custom_date` are missing

The table `ordered_product` stores:

- `customer_zip_code`: the zip_code of the customer that made the order
- `seller_zip_code`: the zip_code of one of the seller sold the product in the order
- `product_id`: the ordered product identifier
- `purchase_day`: the day in which the order has been purchased
- `income`: the sum of the price of each ordered product item
- `quantity`: the number of ordered product items
- `delivery_time`: the difference (in days) between delivered_custom_date delivered_carrier_date

The `Product` table contains product identifier and product category (in English); the table `Location` contains zip code, city, and state for both customers and sellers; and the table `Date` contains all those days such that at least one order has been purchased.

Bonus

Simulate a monthly import from the data source into e-commerce warehouse