The company bigdata2022 is building a new e-commerce website to sell its products across the entire world. In particular the company needs to track all the orders in order to monitor the monthly and annually income. For this reason it has asked us to build a big data solution for their needs to move the old database on the company cluster. In particular the current schema is the following:

- Table orders
    - order_id varchar(255) primary key,
    - customer_id varchar(255),
    - status varchar(255),
    - order_purchase_timestamp varchar(255));
- Table products
    - product_id varchar(255) primary key,
    - product_category_name varchar(255));
- Table items
    - order_id varchar(255),
    - order_item_id varchar(255),
    - product_id varchar(255),
    - price varchar(255));

The new schema that fit company needs is the following:

- Table orders
    - Timestamp of the order
    - User that makes the order (customer_id)
    - List of the ordered products; for each product we want to store id, category, price and the quantity that has been ordered.
    - Total amount of the order
    - Status (Delivered/Cancelled)

- **Task 1**
  Import the ecommerce database into hdfs by using sqoop
- **Task 2**
  Create external tables on the data imported at step 1 and join all the table into a new one named “ordereditems” in csv format
- **Task 3**
  Create a mapreduce job that read the folder of the hive table and stores data into hbase according to orders table description

Design of the hbase table is up to you.

Extra

Build a simple java application or mapreduce jobs that are able to perform the following queries

- Select the total income for a precise month
  public float totalIncomeForMonth(int month);
- For each year select the total income
  public void groupIncomeByYear();
- Select for a particular user the total amount spent for each category
  public void customerOutcomeByCategory(String name, String surname);
- Select the count of orders and the most popular category for each month
  public void analyzeMonths();