תרגיל בית 2 – חלק 1: תיאורטי

מאי 2022

אפרת לוי 301035184

עידן שחמון 315374330

אילן וסילבסקי 322545682

The data:

Users (id, email, language, location)

Transactions (transaction_id, product_id, user_id, purchase_quantity, item_description)

Q1 – Projection: Return the location field

1. Map Function-

```
map (key, value):
    // input - key:  id, value: a line in the Users csv file
    // output - tuple of (location, location)
              for line in value:
                 emit((value[3], value[3]))
```

2. Reduce Function-

```
reduce (key, values):
    // input - key: a location, values: list of locations [location, …, location]
    // output - tuple of the location
        For k in keys:
        emit ((k, k))
```

Q2 - Selection: Return user id with at least one transaction, with purchase quantity greater than 1

1. Map Function-

```
map (key, value):
    // input - key: user_id, value: a line in the Transaction csv file
    // output - tuple of user id and 1
            if line[3]>1:
               emit((user_id, 1))
```

2. Reduce Function-

```
reduce (key, values):
    // input - key: a location, values: list of ones [1, 1, …, 1]
    // output - tuple of the user_id
        For k in keys:
        emit ((k, k))
```

Q3 - Semi-Join: Return users’ detail with at least one transaction

1. Map Function-

```
map (key, value):
    // input – key: file name, value: record of the csv file
    // output - tuple of user id and tuple of table name and the record
        If key == ‘Users’:
                emit(value[0], (“user”,  value))
        If key == ‘Transactions’:
                emit(value[2], (“transaction”, value))
```

2. Reduce Function-

```
reduce (key, values):
    // input – key: user id, value: (tuples from either Users or Transactions)
    // output - tuple of user id and his details
        For value in values:
                        If value[0] == “transaction”:
                                Continue
                        Else:
                                user_details = value[1]
                                For value in values:
                                        If value[0] == “transaction”
                                        Emit(key, user_details)
```

Q4 - Semi-Join: Return users' detail without transaction

1. Map Function-

```
map (key, value):
    // input – key: file name, value: record of the csv file
    // output - tuple of user id and tuple of table name and the record
        If key == 'Users':
                emit(value[0], ("user",  value))
        If key == 'Transactions':
                emit(value[2], ("transaction", value))
```

2. Reduce Function-

```
reduce (key, values):
    // input – key: user id, value: (tuples from either Users or Transactions)
    // output - tuple of user id and his details
        no_tranactions = True
        For value in values:
                        If value[0] == "transaction":
                                no_tranactions = False
                        Else:
                                user_details = value[1]
                        if no_tranactions:
                                Emit(key, user_details)
```

Q5 - Aggregation: Count distinct product purchases for each user (including users without purchases)

1. Map Function-

```
map (key, value):
    // input – key: file name, value: record of the csv file
    // output - tuple of user id null/product id (depending on the source file)
        If key == 'Users':
                emit(value[0], null))
        If key == 'Transactions':
                emit(value[2], value[1])
```

2. Reduce Function-

```
reduce (key, values):
    // input – key: user id, value: list of null and product ids
    // output - tuple of user id and the number of distinct products purchased
        products = []
        for value in values:
                if (value is not null):
                        products.append(value)
        distinct_products = set(products)
        emit ((key, len(distinct_products)))
```