הפקולטה להנדסה
ע״ש איבי ואלדר פליישמן
אוניברסיטת תל אביב

טכנולוגיות ומסדים לנתוני עתק- תרגיל 2

חלק תיאורטי

שאלה 1- projection

return the location field:

```
map (key, val):
// input: key- user id, val- record for this user id (user_id, email, language, location)
// output: tuple of location and 1
        emit((value[location], 1))

reduce (key, values):
// input: key- location, values- list of 1's (unimportant for this section)
// output: tuple of the location (duplicate)
        emit ((key, key))
```

שאלה 2 – selection

return user id with at least 1 transaction with purchase quantity greater than 1:

```
map (key, val):
// input: key- user id, val- record from transaction (transaction_id, product_id, user_id, purchase_quantity, item_description)
// output: tuple of user id and 1
        if val[purchase_quantity]>1:
                emit ((user_id, 1))

reduce (key, values):
// input: key- user id, values- list of 1's (unimportant for this section)
// output: tuple of the user_id (duplicate)
        emit ((key, key))
```

שאלה 3- semi join

return users' detail with at least 1 transaction:

```
map (key, val):
// input: key- table name, val- row from the table (by the key)
// output: tuple of user_id and tuple of table name and the record
    emit((val[user_id], (key,val)))


reduce (key, values):
// input: key- user id, values- list of tuples with (table name, row)
// output: tuple of user_id and his details
    transaction = False
    user_details = False
    for tuple_ in values:
        if tuple_[0] == 'Transactions':
            transaction = True
        else:
            user_details = tuple_[1]
    if transaction & user_details:
        emit ((key, user_details))
```

שאלה 4- semi join

return users' details without transaction:

```
map (key, val):
// input: key- table name, val- row from the table (by the key)
// output: tuple of user_id and tuple of table name and the record
    emit((val[user_id], (key, val)))







reduce (key, values):
// input: key- user id, values- list of tuples with (table name, row)
// output: tuple of user_id and his details
    transaction = False
    user_details = False
    for tuple_ in values:
        if tuple_[0] == 'Transactions':
            transaction = True
        else:
            user_details = tuple_[1]
    if (not transaction) & (user_details):
        emit ((key, user_details))
```

שאלה 5- aggregation

count distinct product purchases for each user (include users without purchases):

```
map (key, val):
// input: key- table name, val- row from the table (by the key)
// output: tuple of user_id and null / product_id (according to the table name)
    if key=='users':
        emit((val[user_id], null))
    else:
        emit((val[user_id], val[product_id]))

reduce (key, values):
// input: key- user id, values- list of null and product_ids
// output: tuple of user_id and the number of products (by the len of the list)
    distinct_products = []
    for i in values:
        if (i is not null) & (i not in distinct_products):
            distinct_products.append(i)
    emit ((key, len(distinct_products)))
```