Martin Rog 1792195

Hussin Almoustafa 1776495

Fares Hamad 1762013

1. **Conversiealgoritme van MongoDB naar Postgresql.**

MongoDB connect = True ,

Products, profiels, sessions = DB collations

#PRODUCTS

Try :

Cur = Postgresql\_connection()

#Function

For item in DB collation{dict, ,keys):

If len(item.keys) > 1: #some data

have just “ \_id” so we ignore it

if len(item.keys) > 7 :

#here we we defined all the

ietms values as variables e.g. ( id = item[“\_id”] )

Cur.execute(“SQL\_INSERT\_INTO\_QUERY table , colm1,colm2 “))

Values(variables)

#That we already defined

Cur.commit

`Else:

#Hier we defined the Unknown values

e.g.(sub\_category = None )

#Not every item have a valueso

we save it as None.

except (Exception, Postgers.DatabaseError) as error:

print(error) #To print if there any sql\_queryerror.

Finally :

Cur = Postgresql\_disconnection(

1. **Filtering algorithm collaborative recommendation persona’s :**

**Postgres\_connection()** #function

**Sql\_select()** #function

**sql\_execute()** #function

**sql\_query( )** #function

Postgres\_connection = True

**select\_profiels\_en\_producten\_from\_****previously\_viewed()**: #function

Sql\_select(“ Select profiels id’s & products id’s from table previously\_viewed and select the name’s of the products and the products targetaudience from table products ” )

Return the result

**select\_profiels\_en\_producten\_from\_orders()**: #function

Sql\_select(“ Select profiels id’s & products id’s from table orders and select the name’s of the products and the products targetaudience from table products ” )

Return the result

#So after that we picked our product’s and the targetaudience we have to filter it.

Volwassenen = [] #a List

Vrouwen = [] #a List

Mannen = [] #a List

Kinderen = [] #a List

gender = ["Baby's", "Jongen", "Meisje", "Volwassenen", "Vrouwen", "Kinderen", "Mannen"] #a List with values of targetaudience.

For every value in the results of both function’s (select\_profiels\_en\_producten\_from\_orders & previously\_viewed) :

If value index[5] == gender index[6] : #then it’s a men product

Mannen.append(value)

elif value index[5] == gender index’s [ 0 or 1 or 5] : #then it’s a child product

Kinderen.append(value)

elif value index[5] == gender index’s[ 2 or 4 ]: #then it’s a women product

Vrouwen.append(value)

#Right now we have 3 list’s with product’s id’s and targetaudience, so we have to make few tables for our recommendation system.

Sql\_query(“create table personas\_mannen “)

Sql\_query(“create table personas\_vrouwen “)

Sql\_query(“create table personas\_kinderen “)

Sql\_query(“create table personas\_recommendations “)

#let’s insert some the products id’s into them.

For every id , targetaudience in our list’s :

Sql\_execute( insert id , targetaudience into the tables)

#so we have almost everything done but we have to decide the id’s of the recommendation’s so we picked 4 id’s from every table [ personas\_mannen , personas\_vrouwen , personas\_kinderen ] and then we insert them into table personas\_recommendations with the targetaudience.

personas\_recommendations() : #function

results = sql\_select(“ select product id , targetaudience from the tables “)

for every item in the results :

sql\_execute(“insert into table personas\_recommendations the results “)

return done 😊

personas\_recommendations()

1. **Content recommendation soortgelijke algorithm :**

sql\_execute #function

sql\_select #function

sql\_query #function

postgres connection = True

sql\_query #Query ( DROP TABLE soortgelijke\_producten )

sql\_query #Query (CREATE TABLE soortgelijke\_producten)

soort\_gelijke(): #function ( This function will select all the diffrents types of subsubcategory and will insert 4 product for evrey subsubcategory)

subsubcategory\_list #List

Result #Query (SELECT subsubcategory FROM products)

For every value in our result:

And we add it to the the list subsubcategory\_list

For every value in subsubcategory\_list:

Producten\_ids #Query ( SELECT id, subsubcategory FROM products )

For every value in producten\_ids:

Prodid is the first index

Subsubcategory is the second index

sql\_execute2 #Query ( INSERT INTO soortgelijke\_producten)