

## 2. Document Store naar Relationale Database

### Onderbouwing

Er is eerst onderzoek gedaan naar de MongoDB en python connectie. Nadat deze werkend was werd dit samengevoegd met de PostgreSQL en python connectie. Hierdoor kon de data uit MongoDB gehaald worden via python, en ook weer via python in PostgreSQL overgezet worden. De try-except 'statement' is gebruikt om velden die af en toe niet aanwezig zijn in de MongoDB op te vullen. De data die geen try-except 'statement' heeft is altijd aanwezig, degene die dat wel hebben zijn dus af en toe niet aanwezig en worden naar null omgezet als dit het geval is. Variabel y is aangemaakt om het uploadproces bij te kunnen houden. Per loop wordt er geprint hoeveel data van het totaal verwerkt is, en ook tussen haakjes het percentage. Vervolgens wordt de data via cur.execute in PostgreSQL gezet. Dit wordt bij products voor elk product gedaan, behalve de 25 die geen naam hebben (vandaar ook continue bij try-except). Bij visitors en sessions worden er 20.000 verwerkt. 2 miljoen zou erg lang duren. Er staat in de code hoelang elke for loop duurt.

### Conversiealgoritme in pseudocode

Connectie met MongoDB maken

Connectie met PostgreSQL maken

Products data uit MongoDB halen

For loop van aantal products

    Variabelen aanmaken per loop

    In die loop data uploaden naar PostgreSQL

Visitors data uit MongoDB halen

Sessions data uit MongoDB halen

For loop die 20000x loopt

    Variabelen aanmaken voor visitors data

    In die loop data uploaden naar PostgreSQL

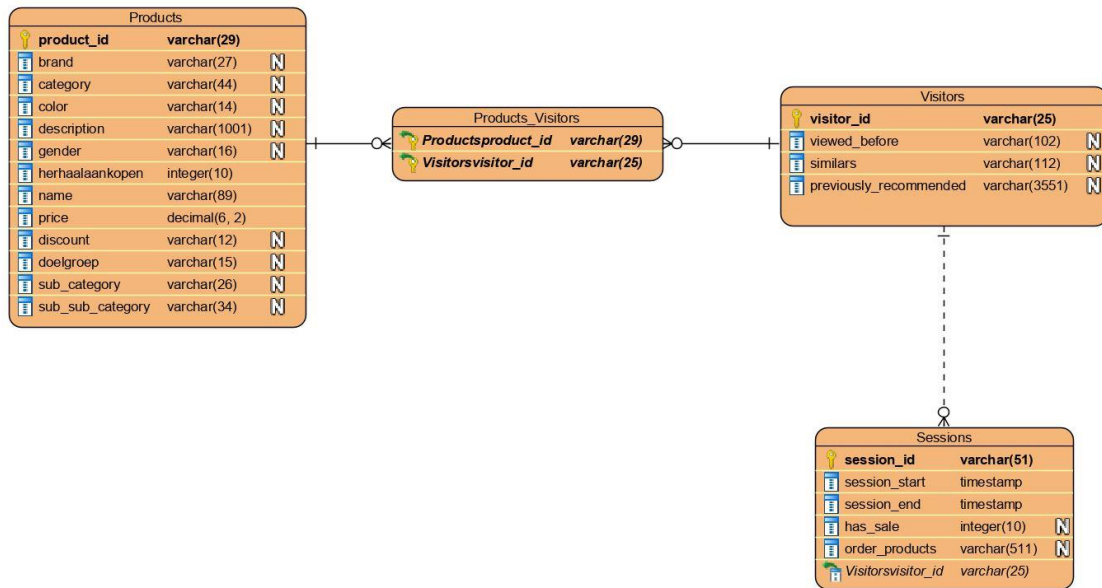
    Variabelen aanmaken voor sessions data

    In die loop data uploaden naar PostgreSQL

Connectie met MongoDB sluiten

Connectie met PostgreSQL sluiten

## Ontwerp van de database



DROP TABLE IF EXISTS products;

DROP TABLE IF EXISTS sessions;

DROP TABLE IF EXISTS visitors;

```
CREATE TABLE products(
product_id VARCHAR(29) NOT NULL,
brand VARCHAR(27) NULL,
category VARCHAR(44) NULL,
color VARCHAR(14) NULL,
description VARCHAR(1001) NULL,
gender VARCHAR(16) NULL,
herhaalaankopen BOOLEAN NOT NULL,
name VARCHAR(89) NOT NULL,
price DECIMAL(6, 2) NOT NULL,
discount VARCHAR(12) NULL,
doelgroep VARCHAR(15) NULL,
sub_category VARCHAR(26) NULL,
sub_sub_category VARCHAR(34) NULL,
PRIMARY KEY(product_id)
```

);

```
CREATE TABLE visitors(  
  visitor_id VARCHAR(25) NOT NULL,  
  viewed_before VARCHAR(102) NULL,  
  similars VARCHAR(112) NULL,  
  previously_recommended VARCHAR(3551) NULL,  
  product_id VARCHAR(29) NULL,  
  PRIMARY KEY(visitor_id),  
  FOREIGN KEY(product_id) REFERENCES products(product_id)  
);
```

```
CREATE TABLE sessions(  
  session_id VARCHAR(51) NOT NULL,  
  session_start TIMESTAMP NOT NULL,  
  session_end TIMESTAMP NOT NULL,  
  has_sale BOOLEAN NOT NULL,  
  order_products VARCHAR(511) NULL,  
  product_id VARCHAR(29) NULL,  
  visitor_id VARCHAR(25) NULL,  
  PRIMARY KEY(session_id),  
  FOREIGN KEY(product_id) REFERENCES products(product_id),  
  FOREIGN KEY(visitor_id) REFERENCES visitors(visitor_id)  
);
```