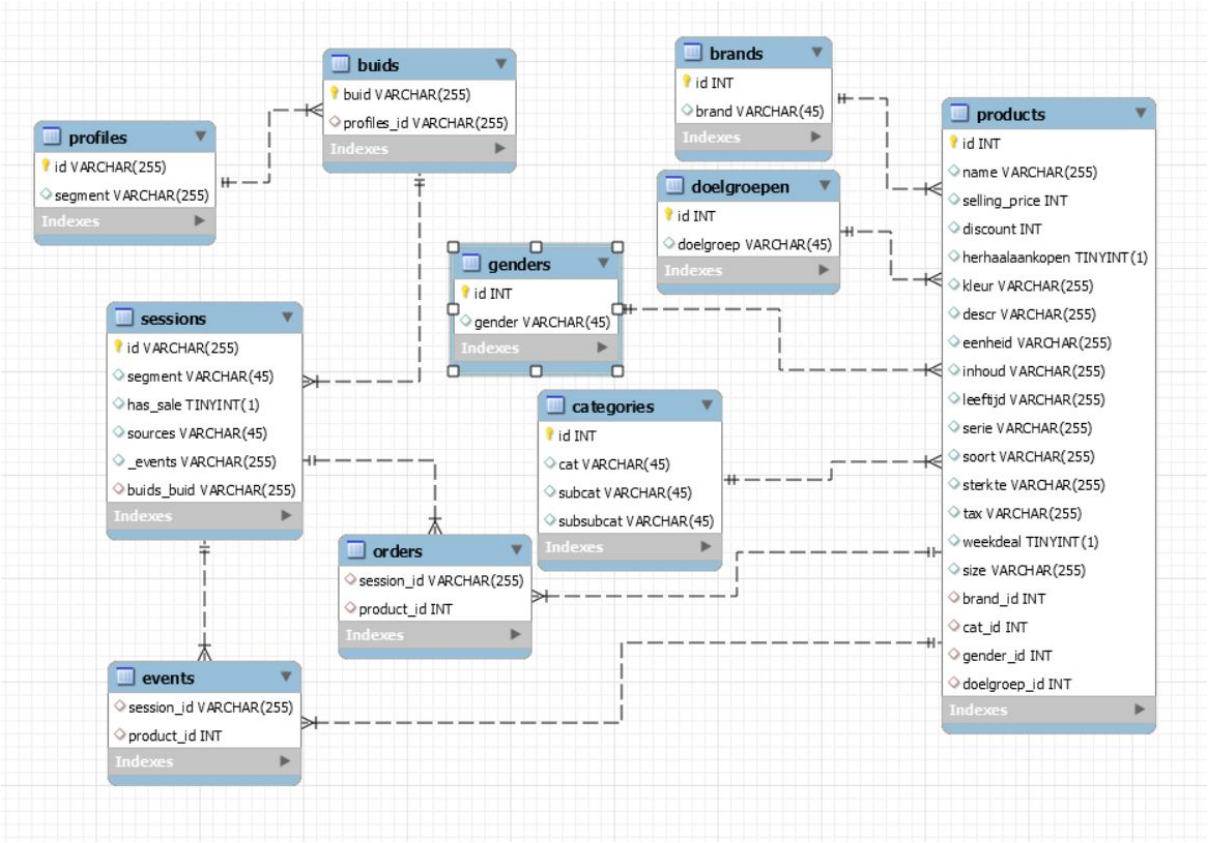
Document Store naar Relationele Database

Structured programming

11-3-2020

Mick Luuring, Michiel Borghuis, Jari Wezer, Jelle Stiesri

Inrichting Database



Keuzes voor data in tabellen: De data die we uiteindelijk hebben gekozen om over te zetten van Mongo naar MySQL is gebaseerd op de bruikbaarheid bij het maken van recommendations. Zo kan het nuttig zijn om te weten in welke maat broek persoon X altijd koopt en minder nuttig of die broek alleen online wordt gekocht. We hebben een lijst gemaakt met alle data die we overslaan en met welke reden. Vaak is de reden dat de informatie van die data altijd null is of dat het irrelevant is, zo zijn afbeeldingen niet interessant voor het maken van een aanbeveling.

Wat vinden wij nuttige data?: Wij hebben rekening gehouden met hoe haalbaar het is om daadwerkelijk aanbevelingen te baseren op de data die we in de database hebben staan. Sommige data zou te ingewikkeld zijn voor ons om te gebruiken en is dus eigenlijk niet zo nuttig voor ons, dit betekend niet automatisch dat het ooit nuttig zou kunnen zijn om recommendations op te baseren. De meest bruikbare data zegt veel over het product en de eigenschappen, vooral over de prijs, formaat, vorige zoekopdrachten en doelgroep.

Tabellen:

Profiles: - **id *** - **segment** **Buids** - **buid *** - **profiles_id **** **Sessions:** - **id *** - **segment** - **has_sale** - **order** - **sources** - **events** - **buids_buid **** **Orders:** - **session_id **** - **product_id **** **Gender** - **id *** - **gender** **Categories:** - **id *** - **cat** - **subcat** - **subsubcat**	**Products:** - **id*** - **name** - **price** - **herhaalaankopen** - **kleur** - **descr** - **discount** - **eenheid** - **inhoud** - **leeftijd** - **serie** - **soort** - **sterkte** - **tax** - **weekdeal** - **size** - **brand_id **** - **cat_id **** - **gender_id **** - **doelgroep_id **** **Brands:** - **id *** - **brand** **Doelgroep** - **id *** - **doelgroep**

Relaties in database:

Profiles > Buids: 1 op veel - 1 profiel kan meerdere buids hebben, 1 buid kan maar 1 profiel hebben

Buids > Sessions: 1 op veel - 1 buid kan meerdere sessies hebben, 1 sessie kan maar 1 buid hebben

Sessions > Orders: 1 op veel - 1 sessie kan meerdere orders hebben, 1 order kan maar 1 sessie hebben

Orders > Products: 1 op veel - 1 order kan maar 1 product hebben, 1 product kan meerdere orders hebben

Products> Categories, Brands, Genders, Doelgroepen: Allemaal 1 op veel- 1 … kan meerdere producten hebben (omdat het ‘normalisatietabellen’ zijn)

Keys:

Primary keys: Alle primary keys zijn ‘id’ omdat in de data die we hebben alle documenten (sessions, products, profiles) al een id hadden, dit is dus makkelijk om ook zo te laten in de MySQL database. De primary keys in de tabellen: brands, gender, doelgroep en categories zijn gebaseerd op welke gender bijvoorbeeld als eerste gevonden word, stel ‘male’ wordt als eerst gevonden, dan zal de id voor ‘male’ 1 zijn. De primarykey van een tabel hebben wij aangegeven met * in bovenstaande tabel.

Foreign key: De primary keys die wij in andere tabellen hebben gebruikt hebben we aangegeven met ** en (naam)_id. Hierdoor kan je goed zien bij welke tabel deze origineel hoord.

Skip Lijst (lijst van data dat we overslaan tijdens het overzetten)

Data	**Rede**
Fast movers	Altijd False
Deeplink	Linkjes ongeldig
Flavour	Altijd null
Images	Irrelevant
Out of stock date	Irrelevant // oude data
Availability	Oude data
Bundle_sku	Altijd null
folder_actief	Irrelevant // altijd null
gebruik	Irrelevant // altijd null
geschiktvoor	Altijd null
geursoort	Irrelevant // altijd null
huidconditie	Altijd null
mid	Irrelevant // altijd null
online_only	Irrelevant // altijd null
shopcart_promo	Irrelevant // altijd null
Session_start	Hetzelfde als segment, dit in combinatie met session_end bepaald de segment. Dus segment is al netjes.
Session_end	Hetzelfde als segment
soorthaar	Altijd null
waterproof	Irrelevant
type	Irrelevant // altijd null
Kleur	Altijd hetzelfde als Color

Bijzonder:

User_agent: We vinden het onethisch om deze data te gebruiken. Je zou met deze info relatief makkelijk achterhalen wie de gebruiker is en welke browsers deze persoon allemaal heeft bezocht enz.

Recommendable: We slaan deze over omdat na het bekijken van de data waarbij recommendable ‘false’ staat we het niet duidelijk vinden waarom we dit niet aan zouden moeten aanraden. We hebben er daarom voor gekozen hier niet naar te kijken.

Stock: de ‘actuele’ voorraad is natuurlijk verouderd. deze data is meer dan een jaar oud en dus niet interessant.

Pseudocode/ code uitleg:

PyMongo:
we maken gebruik van de module ‘pymongo’ om verbinding te maken met de mongoDB. Voor het overzetten van de gefilterde gegevens maken we gebruik van csv, om op een efficiënte manier de relevante data over te zetten in een relationele database.

csvCreator.py:
we slaan de data uit de verschillende collections op en door er slechts 1000 uit te halen per collection voorkomen we dat we erg lang moeten wachten.

csvProducts():
de functie csvProducts neemt de data van de collection products en een filename om hier de data naar toe te schrijven.
We gebruiken een key list om de keys op te slaan van de data die we relevant vinden. Daarnaast gebruiken we nog twee key lists om de keys op te slaan van de data die een laag dieper zit (dictonary in dictionary). Alle data die de code vindt bij de bijbehorende keys wordt opgeslagen in de lijst ‘value_list’.

try except
Omdat de data soms symbolen bevat die python of mysql error’s geeft hebben we ervoor gekozen deze error’s af te vangen en dan de exception te printen.

Forloops en Data structuur aanpassen:
Doordat er in de Mongodatabase veel verschillende nested lists en dictionaries zijn hebben wij ervoor gekozen met forloops door alle informatie die wij relevant vinden te gaan. Dit is om snelheid te besparen. We hebben geen andere manier kunnen vinden om dit efficiënt aan te pakken. Dus, als deze er is zouden we het erg op prijs stellen dit te weten voor het group project over een paar weken.

Problemen die we niet op tijd hebben kunnen oplossen voor de deadline:

SQL foreign key does not exist:
De structuur van de tabellen veranderden naarmate we aan de csv files aan het werken waren. Foreign keys werden primary keys en visa versa. Daardoor kwamen we in de problemen met de csv-files exporteren naar mySQL. We weten wel waar de problemen zitten. De koppeltabel tussen profiles en sessions, ‘buids’, was het probleem. De BUID in de koppeltabel moest de primary key zijn, waarnaar profiles met een foreign BUID-key moest refereren. Wij probeerden de profile ‘id’ te koppelen met de session ‘id’ in de koppeltabel. Dit werkte natuurlijk niet, maar door tijdnood konden we het niet oplossen.