**Implementatie verslag**

**Van : Jason Lie Yung Tjong, 0859607**

**Rinesh Ramadhin, 0882447**

**Module: INFDEV02-5**

**Datum: 28/10/14**

**YouTube Data**

Voor onze case hebben we gekozen om gebruik te van data van Youtube. Deze data importeren we in de Graph database: Neo4j. Doormiddel van Neo4j kunnen data opvragen die gerelateerd is aan de Youtube videos. Hierbij kan worden gedacht aan comments van gebruikers.

**Keuze van data**

Bij het bedenken van deze case hebben we eerst gekeken naar welke data van Youtube beschikbaar is. Youtube heeft twee verschillende API’s beschikbaar gesteld, v.2 en v.3. Youtube API v.2 geeft de gebruiker de mogelijkheid op data op te vragen zonder dat de gebruiker daar API key voor hoeft te gebruiken. Echter geeft deze API maximaal 50 resultaten. Youtube API v.3 heeft geen limiet qua aantal resultaten. Om deze API te gebruiken, moet de gebruiker een API key aanvragen. Per API key heeft de gebruiker 50 miljoen requests per dag.

Uiteindelijk hebben we, ondanks het limiet van resultaten, alleen gebruik gemaakt van Youtube API v.2. Dit komt doordat v.3 gebruik maakt van een unieke Id voor bij het opvragen van een Youtube kanaal. Deze Id bestaat uit 24 willekeurige karakters. Om deze Ids te kunnen gebruiken, hebben we geprobeerd een te generen van 10 miljoen mogelijke combinaties en deze te filteren op werkelijk Ids. Echter duurde dit te lang en hadden na 2 miljoen combinaties nog steeds geen resultaten. Youtube API v.2 geeft daarin tegen de mogelijk om gebruik te maken van de naam van een kanaal om de data op te vragen.

**Data ophalen en verwerken**

Om de data op te halen en te importeren in Neo4j te krijgen, maken we gebruik van Microsoft Visual Studio Professional 2013. Tevens hebben we gebruik gemaakt van de Neo4j console extension die ons de mogelijkheid gaf om de data in Neo4j te bekijken via Visual Studio zelf.

Ons programma geeft de gebruiker de mogelijkheid om de Youtube kanalen uit een tekstbestand te gebruiken of om zelf een kanaal in te voeren waarvan de data zal worden opgehaald. Nadat de gebruiker een kanaal heeft uitgekozen, zal de data van de 50 meest recente video’s van dat kanaal in Neo4j worden geïmporteerd. Doormiddel van cypher queries maakt ons programma eerst de node voor het kanaal aan. Verder pakt hij een video data, creëert hij een node voor de video, en legt hij een relatie tussen de video en het kanaal. De video Id dier eerder was opgehaald, wordt gebruikt in in een API voor de data van comments. Per comment wordt er gekeken naar de inhoud, schrijver en id van de comment. Hiervoor worden weer nodes aangemaakt. De schrijvers van de comments krijgen een aparte nodes zodat meerdere comments aan zelfde schrijver kan worden gerelateerd. Zodra het programma klaar is met de comments, pakt hij de volgende video van het kanaal en herhaald hij het proces.

Als gebruiker kan je de naam van een channel wijzigen en de channel verwijderen (inclusief alle data die hier aan gekoppeld is). Ook is het mogelijk om alle comments van user op te vragen.

**Uitbreiding en beperkingen**

De data die we gebruikt hebben, is maar een klein deel van de data die Youtube beschikbaar stelt. Een voorbeeld hiervan zijn de abonnementen van gebruikers. Youtube geeft de mogelijk om de abonnementen van gebruikers op te vragen, als de gebruiker publiek heeft gemaakt. Hiermee zouden relaties kunnen worden gelegd tussen kanalen en gebruikers. Voor het testen hebben we gebruik gemaakt van 5 verschillende kanalen. Dit is natuurlijk uit te breiden naar een onbeperkt aantal, zolang de gebruiker de namen van de kanalen weet.

Qua beperkingen heeft ons programma dat er maximaal 50 resultaten per keer kunnen worden opgehaald. Dit geldt zowel voor video’s als voor comments. Het is mogelijk om het programma zo te veranderen dat hij alleen de kanalen in het tekst bestand gebruikt en dat hij nadat hij alle kanalen is doorgelopen, opnieuw begint met het eerste kanaal. Hierdoor zullen nieuwe videos en comments continu worden toegevoegd aan de database. Afhankelijk van hoe snel kanalen videos uploaden, zullen alle nieuwe videos worden toegevoegd. Ons programma houdt ook rekening met videos en comments die al in de database staan doormiddel van de Id zodat er geen dubbele nodes in voor komen. Bij comments hangt het ook af van de snelheid dat gebruikers comments plaatsen. Populaire videos zullen sneller meer dan 50 comments krijgen voordat het programma alle kanalen doorlopen heeft. Ook is het zo dat zodra een kanaal een nieuwe video heeft toegevoegd, de comments van de oudste video, niet meer worden toegevoegd.