BEROEPSPRODUCT DDDQ
Casus Top 2000

Geo Bouwmeester - 2159415
Neo Hop - 1662594
Julian van Zwol – 2154913

5 april 2025

HAN_UNIVERSITY
OF APPLIED SCIENCES

INHOUDSOPGAVE

INLEIDING

Databases spelen een cruciale rol in het beheer van grote datasets, vooral in dynamische en jaarlijks terugkerende evenementen zoals de Top 2000. Het ontwerpen van een efficiënte database is essentieel om gegevens over de inzendingen van luisteraars, liedjes, artiesten goed te kunnen opslaan en analyseren.

Dit onderzoeksproject is uitgevoerd door Geo, Neo en Julian als onderdeel van het vak DDDQ, onder de begeleiding van Jorg Janssen. In deze casus hebben wij de bestaande Top 2000-database geanalyseerd en verbeterd. Hierbij gekeken naar de datakwaliteit, het opschonen van de database en het ontwerpen van een efficiënte relationele structuur door middel van verwoordingen en het maken van een CDM/PDM.

1 ONDERZOEK DATAKWALITEIT

1.1 Duiding databasestructuur

De databasestructuur van de Top 2000-database bestaat uit drie tabellen: **Top2000Lijst**, **SongGenre**, en **Song**. Hieronder volgt een beschrijving van de structuur en de onderlinge relaties tussen deze tabellen.

Top2000Lijst, deze tabel bevat de historische ranglijsten van de Top 2000 per editiejaar en bestaat uit de volgende kolommen:

- **editiejaar** (int, not null): Het jaar waarin de Top 2000 editie is gepubliceerd.
- **positie** (int, not null): De positie van het nummer in de lijst.
- **Artiest** (nvarchar(255), null): Naam van de artiest van het nummer.
- **Titel** (nvarchar(255), null): Titel van het nummer.

De tabel heeft editiejaar en positie als gecombineerde primary key, aangezien een positie uniek moet zijn binnen een bepaalde editie.

SongGenre, deze tabel koppelt songs aan hun muziekgenre en bestaat uit de volgende kolommen:

- **artiest** (nvarchar(255), null): Naam van de artiest.
- **titel** (nvarchar(255), null): Titel van de song.
- **genre** (nvarchar(255), null): Het bijbehorende muziekgenre.

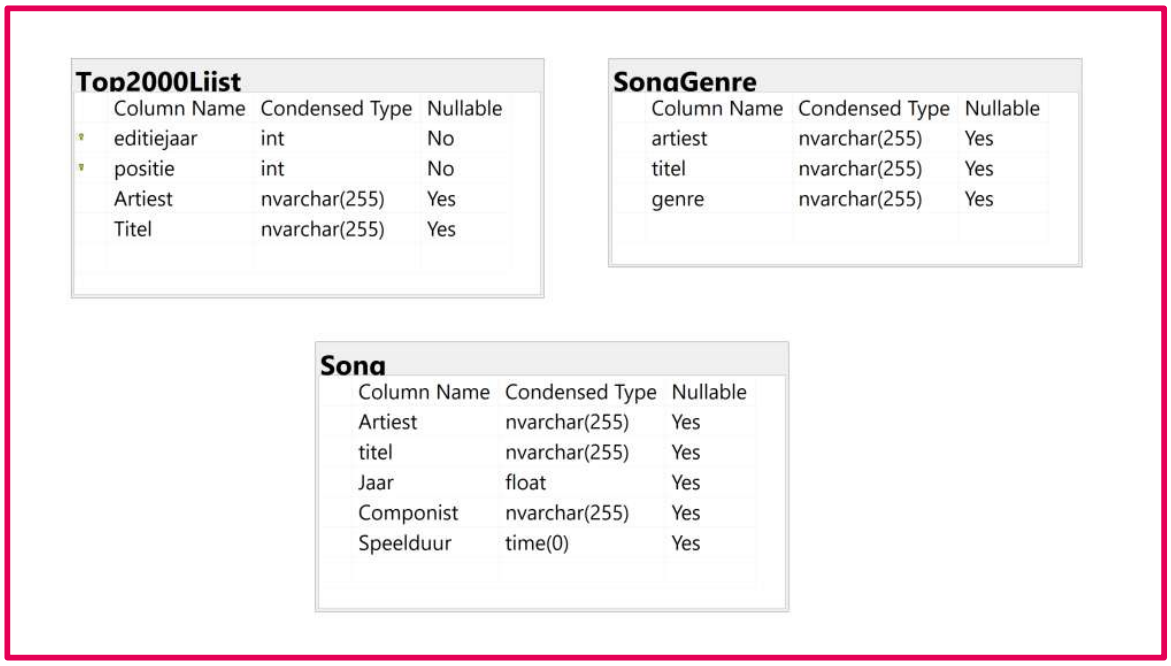
Song, deze tabel bevat aanvullende informatie over een song, zoals het releasejaar, componist en speelduur.

- **Artiest** (nvarchar(255), null): Naam van de artiest.
- **titel** (nvarchar(255), null): Titel van de song.
- **Jaar** (float, null): Het jaar van de release.
- **Componist** (nvarchar(255), null): Naam van de componist(en).
- **Speelduur** (time(0), null): De tijdsduur van het nummer.

Relaties tussen tabellen
Hoewel expliciete foreign keys ontbreken, kunnen we de volgende logische relaties identificeren:

- **Top2000Lijst**(Artiest, Titel) → **Song**(Artiest, titel): Hiermee wordt een nummer in de Top 2000 gekoppeld aan de bijbehorende details in Song.
- **Song**(Artiest, titel) → **SongGenre**(artiest, titel): Hiermee wordt een nummer aan zijn genre gekoppeld.

De database bevat informatie over de Top 2000-lijsten, maar mist formele primary en foreign key constraints. Dit kan leiden tot inconsistenties in de data. In de volgende secties wordt de datakwaliteit nader onderzocht.



Figuur 1: Gegenereerde diagram vanuit SSMS o.b.v. originele database.

1.2 Data extension problemen

Controle op uniforme schrijfwijze

Uitvoering van de SQL select query DKSQ01 in Bijlage A resulteerde in **226** "unieke" genres. Echter hebben sommige genres verschillende schrijfwijzen en daarmee dus geen uniforme manier om genres weer te geven. Enkele voorbeelden van alternatieve schrijfwijzen staan in Bijlage B.

Controle op werkelijkheid

Uitvoering van de SQL select query DKSQ14 in Bijlage B resulteerde in **9** songs waarvan de speelduur de waarde '00:00:00' heeft. Dit kan niet met de werkelijkheid corresponderen.

Controle op schending van referentiele integriteit

Uitvoering van de SQL select query DKSQ02 in Bijlage A resulteerde in schending van de referentiële integriteit op **122** records. Dit houdt in dat **122** songs wel in de Top2000Lijst tabel staan maar niet in de Song tabel.
Uitvoering van de SQL select query DKSQ03 in Bijlage A resulteerde in schending van de referentiële integriteit op **7** records tussen de tabel Song en SongGenre.

Controle op column completeness

Uitvoering van de SQL select query DKSQ04 in Bijlage A resulteerde in een percentage van **96.00%** van nummers in de Top2000 lijst van 2019 waarvan voor elk gegeven attribuut een waarde beschikbaar is. De select queries DKSQ05-07 in Bijlage A zijn gebruikt om de controle uit te voeren per attribuut. De resultaten staan boven de queries in de comments.

Controle op populatie completeness

Uit de SQL select query DKSQ13 in Bijlage A blijkt dat de tabel Top2000Lijst data bevat tot aan editiejaar 2020. Data van recentere jaren ontbreekt dus.

1.3 Data intension problemen

Controle op schending primary key

Voor de controle op schending van de (logische) primary keys is gebruikt gemaakt van een query die records binnen de betreffende tabel groepeert op de (logische) primary key en vervolgens controleert of hier duplicaten tussen zitten. Uitvoering van de SQL select queries DKSQ08-10 in Bijlage A resulteerde in geen schending van de (logische) primary keys van de tabellen: Song, SongGenre en Top2000Lijst.

Redundante data

Het attribuut artiest van alle drie de tabellen verwijst naar dezelfde data. Deze data representeert een persoon en zou om die reden een eigen entiteit moeten zijn. Daarnaast leidt dit tot foutgevoeligheid vanwege het up-to-date houden van dezelfde data op verschillende plekken.

Semantics

Tabel en kolomnamen staan zowel in het Nederlands als Engels. Daarnaast is de casing van de kolomnamen inconsistent. Dit kan leiden tot verwarring en onduidelijkheid over de betekenis. Het kolom Jaar in tabel Song heeft als datatype **float**. Ook dit kan leiden tot verwarring en onduidelijkheid aangezien niet aannemelijk is dat een jaar gedefinieerd zou kunnen worden als een kommagetal.

Controle op derde normaalvorm (3NF)

Volgens de theorie is een tabel in Eerste Normaalvorm (1NF) als elke cel een atomaire waarde bevat. Elke cel voor alle kolommen voor elke tabel bevat een redelijke atomaire (= ondeelbare) waarde, behalve de kolom 'Componist' in tabel 'Song'. Deze bevat een door komma's gescheiden lijst met waarden die de Eerste Normaalvorm schendt. Dit betekent dat de tabel niet genormaliseerd is en zeker niet in Derde Normaalvorm.

Controle op integriteitsregels

IR1. Een song staat maximaal één keer in een editie van de Top 2000.

Uitvoering van de SQL select query [DKSQ11] in Bijlage A resulteerde in schending van de integriteitregel op **2** records. De volgende records werden geïdentificeerd als overtredingen:

- In de editie van 2011 komt Cliff Richard tweemaal voor met Living Doll, op posities 1049 en 1410.
- In de editie van 2020 komt U2 tweemaal voor met One, op posities 170 en 67.

IR2. Elke song die in een editie van de Top 2000 staat, is vóór of ìn het jaar van de betreffende editie uitgebracht.

Uitvoering van de SQL select query [DKSQ11] in Bijlage A resulteerde in geen schending van de integriteitregel.

IR3. De primary key van de tabel Song moet bestaan uit de song titel + hoofdartiest. Dus alle meewerkende artiesten maken geen deel uit van de primary key van Song.

1.4 Data opschonen

Voordat de dataset gebruikt kon worden voor verdere uitbereiding, is het van belang om de oorspronkelijke gegevens uit de Top2000-database op te schonen en te structureren.

In dit hoofdstuk worden de stappen beschreven die zijn genomen om de data om te zetten naar een opgeschoonde versie. Per stap is beschreven hoe de gegevens zijn opgeschoond met bijbehorende SQL-query's die in bijlage D staan. Bijlage D bevat een verkorte versie van het bestand cleanup_queries.sql. Voor een volledig overzicht van alle CREATE TABLE-instructies en constraints kan het oorspronkelijke cleanup_queries.sql-bestand worden geraadpleegd.

1.4.1 Extractie van artiesten

De eerste stap in het opschonen van de dataset bestond uit het extraheren van alle artiestennamen uit de oorspronkelijke Top2000-database. Hierbij is onderscheid gemaakt tussen individuele artiesten die in samenwerking optreden en vaste samenstellingen zoals bands of duo's.

Om dit onderscheid te kunnen maken, zijn er twee query's geschreven. De eerste query [CQSQL1] in bijlage D richtte zich op losse artiestenamen en de artiesten die hebben samengewerkt/een feature zijn van een nummer. Bij nummers met meerdere artiesten werden de namen gescheiden door "&" of "ft.". Deze samengestelde namen zijn opgesplitst en als unieke artiest in de nieuwe tabel gezet.

De tweede query [CQSQL2] in bijlage D haalde vervolgens alle artiesten op die wel als vaste combinatie voorkomen, zoals bands of bekende duo's. Hiervoor is de tabel Artiesten met ampersand.xlsx gebruikt.

1.4.2 Overzetten van nummers

In de tweede stap zijn alle nummers uit de oorspronkelijk database overgezet naar de opgeschoonde database. Hierbij is opnieuw onderscheid gemaakt tussen individuele artiesten (eventueel in samenwerking) en vaste combinaties zoals bands of duo's, zodat alleen de hoofartiest gebruikt word in de identifier van een song.
De eerste query [CQSQL3] in bijlage D heeft alle losse artiesten en nummers met meerdere artiesten overgezet.
De tweede query [CQSQL4] in bijlage D heeft alle bands/duo's afgehandeld.

1.4.3 Toevoegen van featured artiesten

Na het overzetten van de hoofdartiesten, was de volgende stap het expliciet vastleggen van de featured artiesten. Deze data word opgenomen in een aanvullende tabel: SongFeaturedArtist.

Doormiddel van query [CQSQL5] in bijlage D zijn alle featured artiesten geidentificeerd uit de oorsponkelijke data en overgezet.

1.4.4 Splitsen en koppelen van componisten

In deze stap is de focus gelegd op het verwerken van de componisteninformatie. In de oorspronkelijke dataset werd de naam of namen van de componist(en) opgeslagen als één gecombineerde tekstwaarde, waarbij meerdere namen vaak met een komma van elkaar gescheiden waren. Om deze informatie beter bruikbaar te maken voor analyse en verdere verrijking, zijn de componisten eerst als losse entiteiten opgeslagen in een aparte tabel, en vervolgens gekoppeld aan de bijbehorende nummers.

Met de eerste query [CQSQL6] in bijlage D zijn alle unieke componistnamen overgezet. En met de tweede query [CQSQL7] in bijlage D zijn alle componisten weer gekoppeld aan het juiste nummer.

1.4.5 Overzetten en opschonen van genres

Voor stap 5 zijn alle genres voor elk nummer overgezet en dubbele genre namen die net anders zijn getypt of vertaald gecorrigeerd.
Voor het overzetten van alle genres voor elke song zijn de volgende twee query's [CQSQL8] en [CQSQL9] in bijlage D.

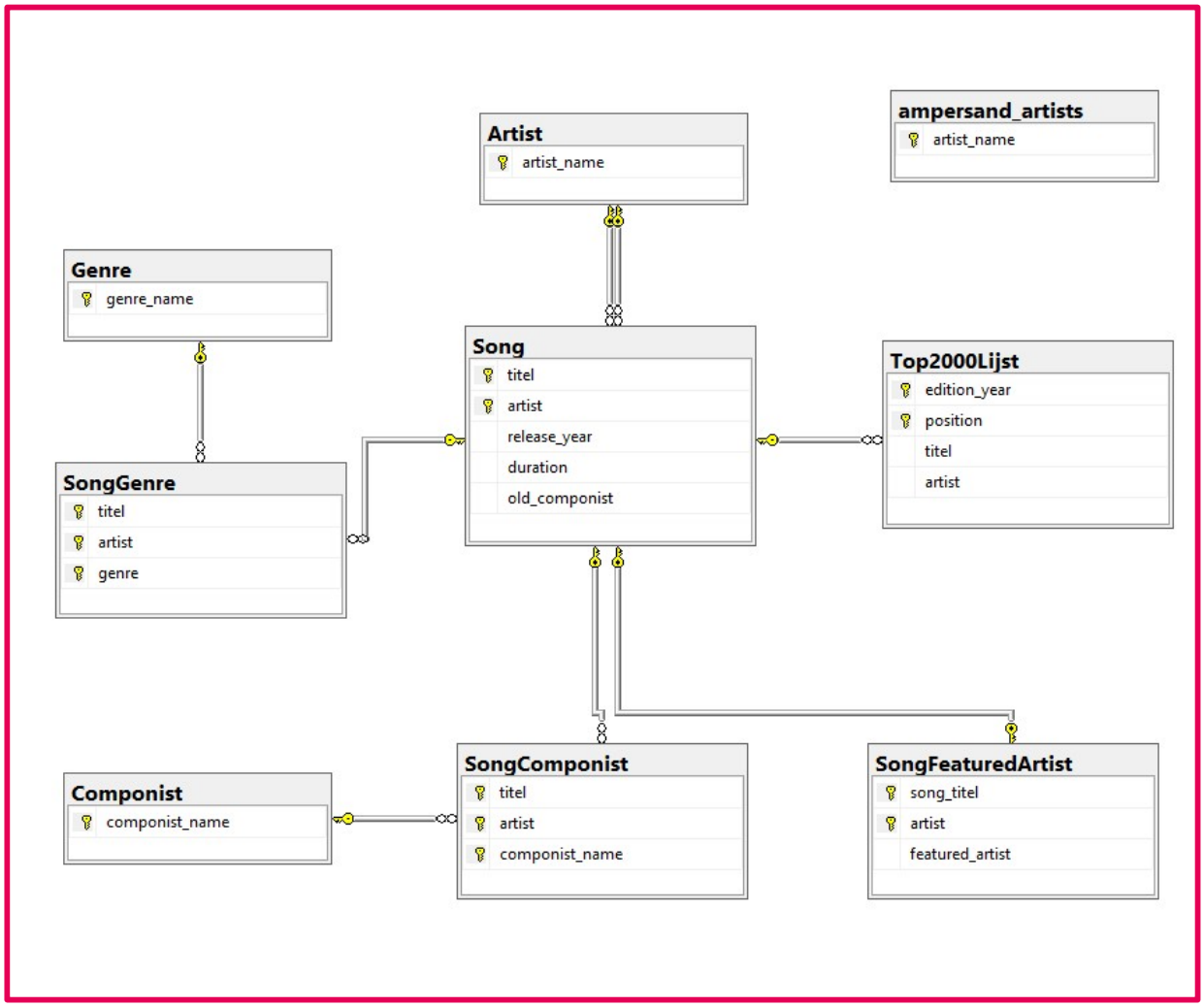
Na het overzetten van de genre-informatie zijn meerdere correcties uitgevoerd om dubbele of foutief gespelde genrenamen te corrigeren [CQSQL10] t/m [CQSQL15] in bijlage D. Denk hierbij aan het corrigeren van inconsistenties zoals "allternatieve rock" naar "alternatieve rock".
Ten slotte zijn alle unieke genre namen in een aparte tabel gezet met de query [CQSQL16] in bijlage D.

1.4.6 Overzetten van Top2000-lijsten

In de laatste stap zij alle edities van de Top2000-lijsten overgezet. Voor elk jaar werd de positie van elk nummer vastgelegd, gekoppeld aan de juiste combinatie van titel en hoofdartiest. Hiervoor zijn de query's [CQSQL17] en [CQSQL18] in bijlage D gebruikt.

1.4.7 Resulterende database

De cleanup_queries.sql resulteerd in de volgende database diagram.



2 VERBETERDE VERSIE

2.1 Feitenverwoording en analyse

Tijdens het opstellen van de casus hebben we voor opdracht 2 en 3 verschillende verwoordingen gecreëerd, geanalyseerd en gedetailleerd beschreven in bijlage D.

2.2 Keuze modelleerwijze

FT3
Bij FT3 is ervoor gekozen om genre op te slaan als een entiteit in plaats van attribuut. De reden hiervoor is zodat van genre een domein tabel gemaakt kan worden welke makkelijk in een dropdown lijstje in de GUI gebruikt kan worden. Bovendien kan er op deze manier makkelijker gevalideerd worden. Enkel waarden in de domein tabel zijn toegestaan. Zo voorkomen we foute of niet-gedefinieerde invoer.

FT6
Bij FT6 is het editie jaar van de Top2000 en het nummer gekoppeld in één entiteit. Dit voorkomt dat song twee keer per editie toegevoegd kan worden. Deze business rule wordt nu dus al in het model afgevangen. Er ontstaat nu wel een business rule dat iedere editie niet twee keer dezelfde positie kan hebben. Dit komt omdat positie niet meegenomen wordt in de primaire key. Dit moet dus nog afgevangen worden met BR1.

Het problem bij FT6 kent dus ook verschillende modelleerwijze. Eén waarbij editiejaar en song in de primaire key worden opgenomen (huidige oplossing) en de alternatieve wijze is editiejaar en positie in de primaire key opnemen. De entiteit Top2000ListEntry krijgt dan niet een afhankelijke relatie naar song maar een een op meer relatie waarbij BR1 afgevangen moet worden.

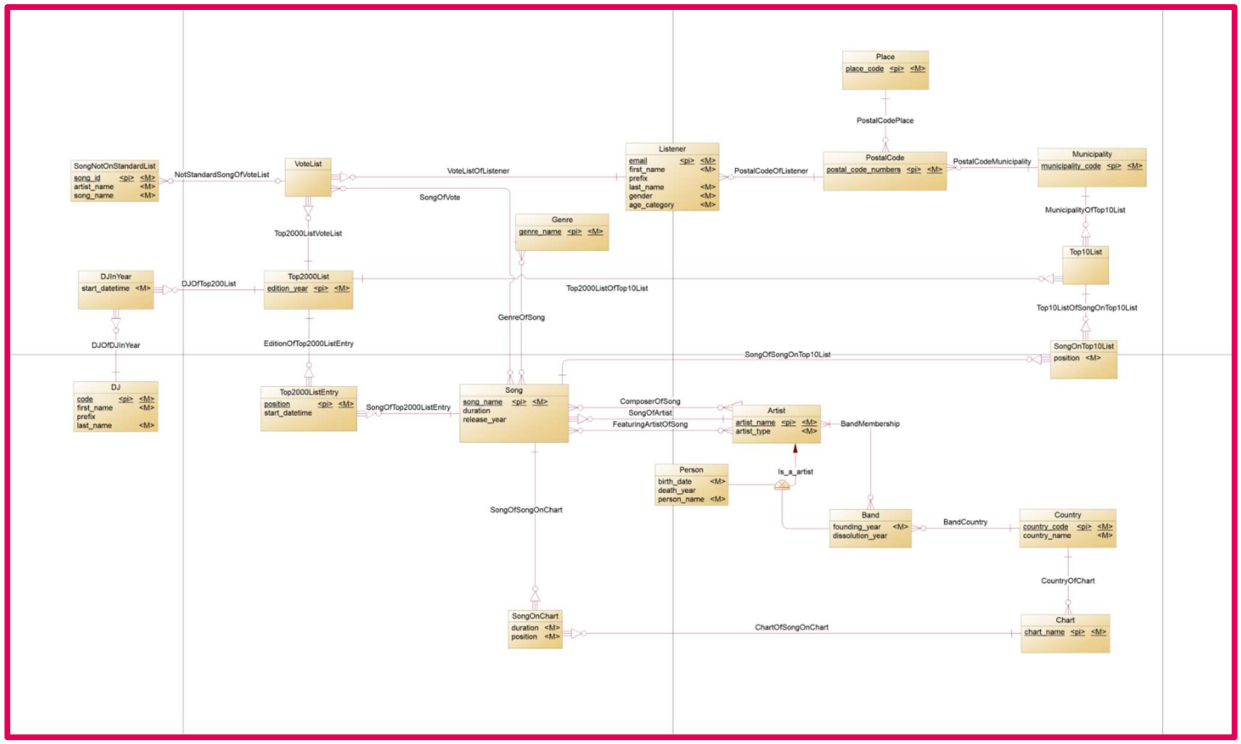
FT11
age_category slaan we niet op als geboortedatum waarmee via programmatuur de categorie berekend kan worden maar daadwerkelijk als categorie. De reden hiervoor is zodat er niet te veel persoonlijke informatie gevraagd wordt van een luisteraar het de drempel om te stemmen daarmee verlaagd wordt.

FT14
SongNotOnStandardList kan niet geidentificeerd worden door song_name en artist_name aangezien dit vrije invoer velden zijn voor een luisteraar. We kunnen er dus niet op vertrouwen dat deze altijd correct worden ingevuld. Bovendien kunnen verschillende luisteraars hetzelfde song_name en artist_name opgeven van een song niet uit de standaard lijst. Vandaar dat er een ID property is toegevoegd.

FT15, FT16
FT15 is toegevoegd als feit type om aan te geven dat een artiest van verschillende type kan zijn. FT16 geeft aan dat artiest van type band kan zijn. In dat geval krijgt hij dus ook o.a. founding_year. Een waarde die natuurlijk niet relevant is voor bijvoorbeeld Persoon een ander subtype van Artiest. Omdat geboorte datum dan weer niet relevant is voor band is dit attribuut alleen toegevoegd aan subtype Persoon. Omdat zowel band als person wel gekenmerkt worden door hun artiesten naam staat dit attribuut en primaire identifier in het supertype Artiest.

FT17
Entiteit Country wordt geïdentificeerd met een landcode omdat dit nog steeds voldoende context geeft, makkelijk weer te geven is in een GUI. Dit leidt ook tot minder opslag omdat entiteit Band niet een volledige naam (nvarchar(255)) maar enkel de code (char(2)) hoeft op te slaan.

2.3 CDM



Figuur 2: CDM van vernieuwde Top2000 Database

2.4 Declaratieve constraints.

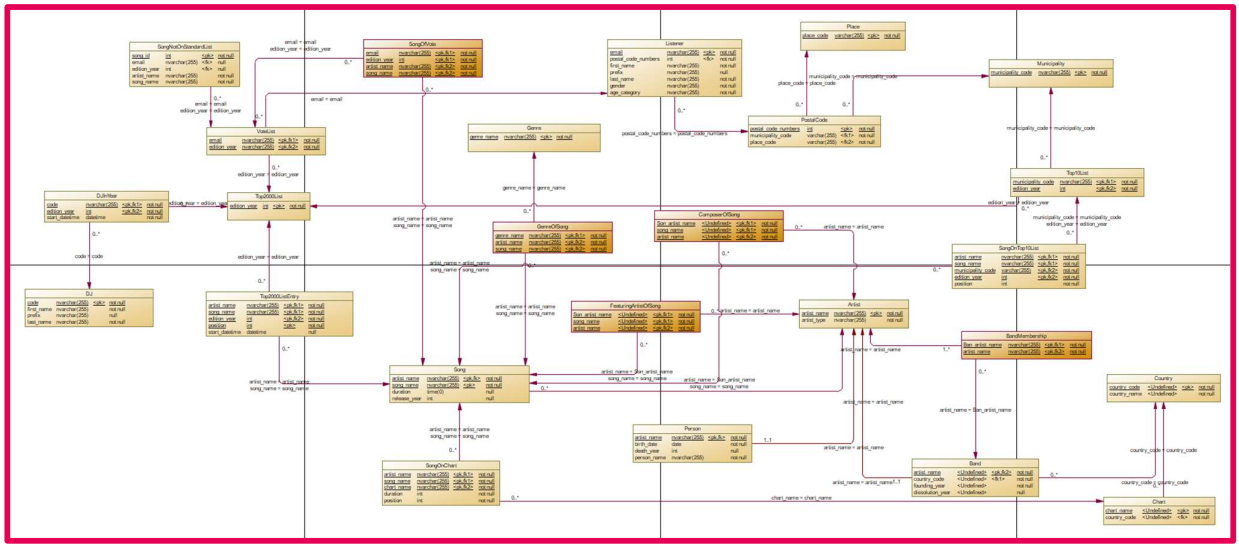
Wanneer we kijken naar onze integriteitsregels die niet automatisch worden afgevangen binnen het CDM, kunnen we een lijst opstellen van bijbehorende constraints, oftewel business rules. In de toekomst zouden deze business rules als declaratieve constraints opgenomen kunnen worden in het DLL-script.

BR1 Twee verschillende nummers mogen niet dezelfde positie innemen binnen dezelfde lijst (editiejaar).
BR2 Een Top 2000-lijst bevat altijd exact 2000 unieke nummers.
BR3 Een stemlijst, met de voorkeuren van een luisteraar, bevat minimaal 5 en maximaal 35 songs.
BR4 De postcode van een luisteraar bestaat altijd uit precies 4 cijfers.
BR5 Twee verschillende nummers mogen niet dezelfde positie innemen binnen dezelfde top 10 lijst per gemeente (editiejaar).
BR6 Een Top 10-lijst bevat altijd exact 10 unieke nummers.
BR7 Nadat de Top 2000 van een jaar is uitgezonden, worden de bijbehorende stemmen van luisteraars verwijderd.
BR8 Een vrije keuze song (SongNotOnStandardList) van een luisteraar mag slechts één keer per editiejaar in zijn of haar stemlijst voorkomen.
BR9 Een tijdsblok van een dj dient altijd exact 2 uur te beslaan.
BR10 Een DJ mag meerdere keren draaien, maar niet op hetzelfde moment.
BR11 De starttijd van een DJ moet altijd later zijn dan de starttijd van de vorige DJ op dezelfde dag.

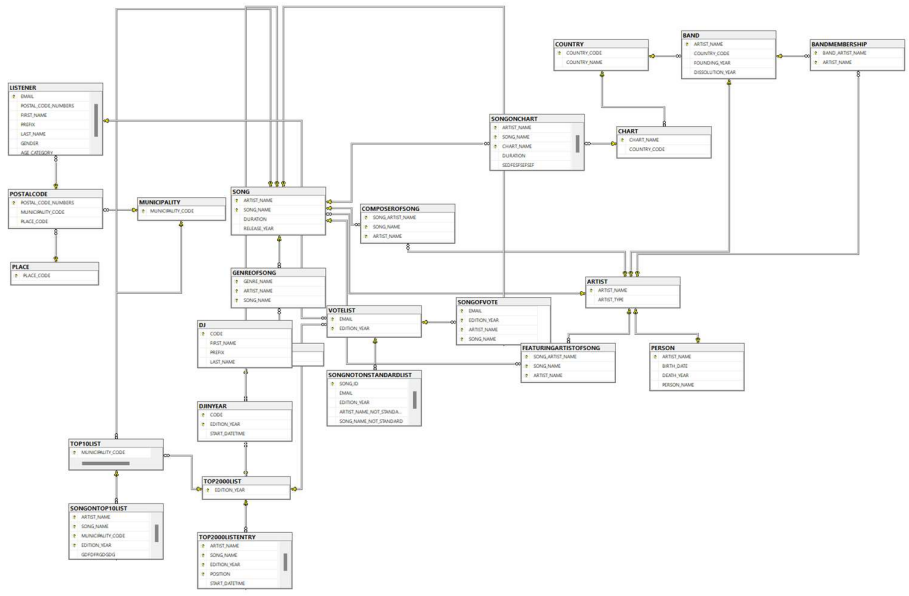
BR12 De waarde van highest_ranking_position moet een positief geheel getal zijn.
BR13 De waarde van position (positie van een nummer in de lijst) moet een positief geheel getal zijn.
BR14 Het attribuut percentage_of_all_votes moet tussen 0 en 100 liggen.

BR15 De duration van een nummer in een hitlijst moet een positief geheel getal zijn (aantal weken).
BR16 De waarde van release_year moet een geldig jaartal zijn, bijvoorbeeld tussen 1900 en het huidige jaar.
BR17 De founding_year van een band moet een geldig jaartal zijn, bijvoorbeeld tussen 1900 en het huidige jaar.
BR18 De birth_date van een persoon vóór de huidige datum liggen.
BR19 De death_year van een persoon moet vóór de huidige datum liggen, tenzij de persoon nog in leven is.
BR20 De waarde van duration moet een positief geheel getal zijn, dat de duur van een nummer in het formaat uren:minuten:seconden weergeeft.

2.5 PDM



Figuur 3: PDM van vernieuwde Top2000 database



2.6 Database testen.

Voor het testen van de nieuwe database is eerst alle data van de opgeschoonde database over gezet naar de uiteindelijke database. Zie Bijlage F.

We hebben een SQL-query opgesteld om de gegevens van het jaar 2017 weer te geven, inclusief de volgende kolommen: NR., VERLOOP, TITEL, ARTIEST en JAAR. Voor de SQL verwijs ik u graag naar Bijlage E.

	NR	VERLOOP	TITEL	ARTIEST	JAAR
1	1	0	Bohemian Rhapsody	Queen	1975
2	2	0	Hotel California	Eagles	1977
3	3	0	Stairway to Heaven	Led Zeppelin	1971
4	4	0	Piano Man	Billy Joel	1974
5	5	0	Child in Time	Deep Purple	1972
6	6	4	Black	Pearl Jam	1993
7	7	2	Wish You Were Here	Pink Floyd	1975
8	8	3	Fix You	Coldplay	2005
9	9	-3	Avond	Boudewijn de Groot	1997
10	10	7	November Rain	Guns N' Roses	1992
11	11	-4	Heroes	David Bowie	1977
12	12	4	Comfortably Numb	Pink Floyd	1979
13	13	8	Thunderstruck	AC/DC	1990
14	14	-6	Mag ik dan bij jou	Claudia de Breij	2011
15	15	0	Nothing Else Matters	Metallica	1992
16	16	-4	Imagine	John Lennon	1971
17	17	2	Brothers in Arms	Dire Straits	1985
18	18	-5	Purple Rain	Prince & The Revolution	1984
19	19	-5	The River	Bruce Springsteen	1981
20	20	8	Hurt	Johnny Cash	2002
21	21	1	Radar Love	Golden Earring	1973
22	22	-2	Shine On You Crazy Diamond	Pink Floyd	1975
23	23	2	One	Metallica	1994

Figuur 4: Lijst van top 10 songs uit 2017

Voor de Top 10 per gemeente is de query [ESQL2] in bijlage E geschreven. Deze query geeft een top 10 per gemeente met totaal aantal stemmen per song en het percentage aan stemmen per song.

	ARTIST_NAME	SONG_NAME	aantalstemmen	percentage_van_totaal
1	Amy Winehouse	Back to black	4	3.360000000000
2	Coldplay	Fix You	4	3.360000000000
3	AC/DC	Thunderstruck	4	3.360000000000
4	Red Hot Chili Peppers	Under the bridge	3	2.520000000000
5	Nirvana	Smells like teen spirit	3	2.520000000000
6	Fleetwood Mac	Go your own way	3	2.520000000000
7	David Bowie	Heroes	3	2.520000000000
8	Queen	Bohemian Rhapsody	3	2.520000000000
9	Phil Collins	In the air tonight	3	2.520000000000
10	Metallica	Nothing Else Matters	3	2.520000000000

Figuur 5: Top 10 per gemeente

3 LIJST VAN FIGUREN

4 LIJST VAN TABELLEN

BIJLAGE A.

```sql
-- [DKSQ01] Geen uniforme schrijfwijze
SELECT DISTINCT genre
FROM [Top2000].[dbo].[SongGenre]

-- [DKSQ02] Check on Top2000Lijst's (logische) referentie naar Song
SELECT Artiest, Titel
FROM Top2000Lijst as list
WHERE NOT EXISTS (
      SELECT *
      FROM Song as song
      WHERE list.Artiest = song.Artiest
      AND list.Titel = song.titel
)

-- [DKSQ03] Check on SongGenre's (logische) referentie naar Song
SELECT artiest, titel
FROM SongGenre as genre
WHERE NOT EXISTS (
      SELECT *
      FROM Song as song
      WHERE genre.Artiest = song.Artiest
      AND genre.Titel = song.titel
)

-- [DKSQ04] Check on value for every given property for songs in the Top2000 list
in 2019.
SELECT (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Top2000Lijst WHERE editiejaar =
'2019')) as completeness_percentage
FROM (
            SELECT song.Artiest,
                  song.titel,
                  song.Jaar,
                  song.Speelduur,
                  song.Componist,
                  list.editiejaar,
                  COUNT(genre.genre) AS number_of_genres
            FROM Top2000Lijst AS list
            INNER JOIN Song AS song
            ON list.Artiest = song.Artiest
            AND list.Titel = song.titel
            LEFT JOIN SongGenre as genre
            ON genre.artiest = song.Artiest
            AND genre.titel = song.titel
            WHERE list.editiejaar = '2019'
            GROUP BY song.Artiest,
                  song.titel,
                  song.Jaar,
                  song.Speelduur,
                  song.Componist,
                  list.editiejaar,
                  genre.artiest,
                  genre.titel
            HAVING COUNT(genre.genre) >= 1
            AND song.Componist IS NOT NULL
            AND song.Speelduur IS NOT NULL
) as Dataset
```

```sql
-- [DKSQ05] 100% of the songs in the Top2000 list in 2019 has a value for
duration
SELECT (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Top2000Lijst WHERE editiejaar =
'2019')) as completeness_percentage
FROM (
            SELECT song.Artiest,
                  song.titel,
                  song.Jaar,
                  song.Speelduur,
                  song.Componist,
                  list.editiejaar,
                  COUNT(genre.genre) AS number_of_genres
            FROM Top2000Lijst AS list
            INNER JOIN Song AS song
            ON list.Artiest = song.Artiest
            AND list.Titel = song.titel
            LEFT JOIN SongGenre as genre
            ON genre.artiest = song.Artiest
            AND genre.titel = song.titel
            WHERE list.editiejaar = '2019'
            GROUP BY song.Artiest,
                  song.titel,
                  song.Jaar,
                  song.Speelduur,
                  song.Componist,
                  list.editiejaar,
                  genre.artiest,
                  genre.titel
            HAVING song.Speelduur IS NOT NULL
) as Dataset

-- [DKSQ06] 98.60% of the songs in the Top2000 list in 2019 has a value for
composer
SELECT (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Top2000Lijst WHERE editiejaar =
'2019')) as completeness_percentage
FROM (
            SELECT song.Artiest,
                  song.titel,
                  song.Jaar,
                  song.Speelduur,
                  song.Componist,
                  list.editiejaar,
                  COUNT(genre.genre) AS number_of_genres
            FROM Top2000Lijst AS list
            INNER JOIN Song AS song
            ON list.Artiest = song.Artiest
            AND list.Titel = song.titel
            LEFT JOIN SongGenre as genre
            ON genre.artiest = song.Artiest
            AND genre.titel = song.titel
            WHERE list.editiejaar = '2019'
            GROUP BY song.Artiest,
                  song.titel,
```

```
                song.Jaar,
                song.Speelduur,
                song.Componist,
                list.editiejaar,
                genre.artiest,
                genre.titel
            HAVING song.Componist IS NOT NULL
) as Dataset

-- [DKSQ07] 97.25% of the songs in the Top2000 list in 2019 has a value for genre
SELECT (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Top2000Lijst WHERE editiejaar =
'2019')) as completeness_percentage
FROM (
            SELECT song.Artiest,
                song.titel,
                song.Jaar,
                song.Speelduur,
                song.Componist,
                list.editiejaar,
                COUNT(genre.genre) AS number_of_genres
            FROM Top2000Lijst AS list
            INNER JOIN Song AS song
            ON list.Artiest = song.Artiest
            AND list.Titel = song.titel
            LEFT JOIN SongGenre as genre
            ON genre.artiest = song.Artiest
            AND genre.titel = song.titel
            WHERE list.editiejaar = '2019'
            GROUP BY song.Artiest,
                song.titel,
                song.Jaar,
                song.Speelduur,
                song.Componist,
                list.editiejaar,
                genre.artiest,
                genre.titel
            HAVING COUNT(genre.genre) >= 1
) as Dataset


-- [DKSQ08] Check op tabel Top2000Lijst.
SELECT editiejaar, positie
FROM Top2000Lijst
GROUP BY editiejaar, positie
HAVING COUNT(*) <> 1


-- [DKSQ09] Check op tabel Song.
SELECT Artiest, titel
FROM Song
GROUP BY Artiest, titel
HAVING COUNT(*) > 1


-- [DKSQ10] Check op tabel SongGenre.
```

```
SELECT artiest, titel, genre
FROM SongGenre
GROUP BY artiest, titel, genre
HAVING COUNT(*) > 1


-- [DKSQ11] IR1. Een song staat maximaal een keer in een editie van de Top 2000.
SELECT editiejaar, Artiest, Titel, COUNT(*) AS aantal,
STRING_AGG(CAST(positie AS VARCHAR(10)), ', ') AS posities
FROM Top2000Lijst
GROUP BY editiejaar, Artiest, Titel
HAVING COUNT(*) > 1
ORDER BY editiejaar, aantal DESC;

-- [DKSQ12] IR2. Elke song die in een editie van de Top 2000 staat, is voor of in
het jaar van de betreffende editie uitgebracht.
SELECT song.Artiest,
      song.titel,
      song.Jaar,
      list.editiejaar
FROM Top2000Lijst AS list
INNER JOIN Song AS song
ON list.Artiest = song.Artiest
AND list.Titel = song.titel
WHERE song.Jaar > list.editiejaar


-- [DKSQ13] Check on population completeness
SELECT [editiejaar], count(*)
FROM [Top2000].[dbo].[Top2000Lijst]
GROUP BY editiejaar
ORDER BY editiejaar desc

-- [DKSQ14] Check on accuracy
SELECT TOP (1000) [Artiest]
      ,[titel]
      ,[Jaar]
      ,[Componist]
      ,[Speelduur]
  FROM [Top2000].[dbo].[Song]
  where [Speelduur] = '00:00:00'
```

BIJLAGE B.

Tabel 1: Alternatieve schrijfwijzes

Optie 1	Optie 2	Optie 3
alternatieve rock	alternative rock	-
elektronisch	elektronische muziek	elketronische muziek
hard rock	hardrock	-
Keltisch	Keltische muziek	-
kerstlied	kerstmuziek	-
klassiek	klassieke muziek	-
poppunk	popunk	-

psychedelic rock	psychedelische rock	-
rock-'n -roll	rock-'n-roll	-

Tabel 2: Songs met een speelduur van 00:00:00

Artiest	Titel	Jaar	Componist	Speelduur
André Hazes	Geef mij je angst	1984	Udo Jürgens, Michael Kunze, André Hazes	00:00:00
BLØF & Counting Crows	Holiday in Spain	2004	NULL	00:00:00
Ella Fitzgerald & Louis Armstrong	Summertime	1957	George Gershwin, DuBose Heyward	00:00:00
Elton John	Circle of Life	1994	Elton John, Tim Rice	00:00:00
Guus Meeuwis	Geef mij je angst	2004	Udo Jürgens, Michael Kunze, André Hazes	00:00:00
Mike Oldfield	Tubular Bells	1974	Mike Oldfield	00:00:00
Nena	99 Luftballons	1983	Carlo Karges, Uwe Fahrenkrog-Petersen	00:00:00
Nielson	IJskoud	2018	Niels Littooij, Don Zwaaneveld, Lodewijk Martens	00:00:00
Sammy Davis jr.	Mr. Bojangles	1972	Jerry Jeff Walker	00:00:00

BIJLAGE C.

FT1

```
Bohemian Rhapsody   van Queen          is uitgebracht in 1975.
Somebody To Love    van Queen          is uitgebracht in 1976.
Somebody To Love    van George Michael is uitgebracht in 1993.
Imagine             van John Lennon    is uitgebracht in 1971.
All For Love        van Bryan Adams    is uitgebracht in 1994.
ET: Song                                                att release_year
ID: att song_name + ET Artist
                     ID: artist_name
RT: SongOfArtist between ET Song (dep) and ET Artist
Predicate: <song_name> van <artist_name> is uitgebracht in <release_year>.
```

FT2

```
Bohemian Rhapsody   van Queen    duurt 00:05:57.
Changes             van 2Pac     duurt 00:04:30.
ET: Song                              att duration
MATCH
Predicate: <song_name> van <artist_name> duurt <duration>
```

FT3

```
In the end                van Linkin Park     heeft als genre nu metal
In the end                van Linkin Park     heeft als genre rapcore
Let it be                 van The Beatles     heeft als genre rock
With or without you       van U2              heeft als genre rock
ET: Song                                                ET: Genre
MATCH                                                   ID: att genre_name
RL: GenreOfSong between ET Song and ET Genre
Predicate : <song_name> van <artist_name> heeft als genre <genre_name>.
```

FT4

```
Aan Somebody To Love      van George Michael  wordt meegewerkt door Queen.
Aan All For Love          van Bryan Adams     wordt meegewerkt door Rod Stewart.
Aan All For Love          van Bryan Adams     wordt meegewerkt door Sting.
ET: Song                                                                ET: Artist
MATCH                                                                   MATCH
RT: FeaturingArtistOfSong between ET Song and ET Artist
Predicate: Aan <song_name> van <artist_name> wordt meegewerkt door <artist_name>.
```

FT5

```
Somebody To Love    van George Michael  is gecomponeerd     door Freddie Mercury.
Summer of '69       van Bryan Adams     is componeerd       door Bryan Adam
```

```
Summer of '69       van Bryan Adams     is componeerd       door Jim Vallance
ET: Song                                                    ET: Artist
MATCH                                                       MATCH
RT: ComposerOfSong between ET Song and ET Artist
Predicate: <song_name> van <artist_name> is componeerd door <artist_name>.
```

FT6

```
In de lijst van 2017 staat Bohemian Rhapsody        van Queen                op nummer 1.
In de lijst van 2017 staat Imagine                  van John Lennon          op nummer 16.
In de lijst van 2017 staat The Sound Of Silence     van Simon & Garfunkel    op nummer 28.
In de lijst van 2017 staat Somebody To Love         van Queen                op nummer 66.
In de lijst van 2017 staat Somebody To Love         van George Michael       op nummer 135.
In de lijst van 2015 staat Imagine                  van John Lennon          op nummer 1.
In de lijst van 2015 staat Bohemian Rhapsody        van Queen                op nummer 2.
In de lijst van 2015 staat Somebody To Love         van Queen                op nummer 65.
In de lijst van 2015 staat Somebody To Love         van George Michael       op nummer 517.
In de lijst van 2013 staat All For Love             van Bryan Adams          op nummer 1987.
ET: Top2000ListEntry                                                            att position
ID: ET Top2000List  +      ET Song
    ID: att edition_year  MATCH
RT: EditionOfTop2000ListEntry between ET Top2000ListEntry (dep) and ET Top2000List
RT: SongOfTop2000ListEntry between ET Top2000ListEntry (dep) and ET Song
Predicate: In de lijst van <edition_year> staat <song_name> van <artist_name> op nummer <position>.
```

FT7

```
De luisteraar met email marco.engelbart@hotmail.com heeft voornaam Marco.
De luisteraar met email jan.van.vliet@gmail.com     heeft voornaam Jan.
ET Listener                                                        att first_name
ID: att email
Predicate: De luisteraar met email <email> heeft voornaam <first_name>.
```

FT8

```
De luisteraar met email jan.van.vliet@gmail.com   heeft tussenvoegsel van.
De luisteraar met email piet.de.bruin@hotmail.com heeft tussenvoegsel de.
ET Listener                                                          att prefix
MATCH
Predicate: De luisteraar met email <email> heeft tussenvoegsel <prefix>.
```

FT9

```
De luisteraar met email marco.engelbart@hotmail.com heeft achternaam Engelbart.
De luisteraar met email jan.van.vliet@gmail.com     heeft achternaam Vliet.
ET Listener                                                        att last_name
MATCH
```

```
Predicate: De luisteraar met email <email> heeft achternaam <last_name>.
```

FT10

```
De luisteraar met email marco.engelbart@hotmail.com heeft geslacht man.
De luisteraar met email jan.van.vliet@gmail.com     heeft geslacht vrouw.
ET Listener                                                        att gender
MATCH
Predicate: De luisteraar met email <email> heeft <gender>.
```

FT11

```
De luisteraar met email marco.engelbart@hotmail.com valt in de leeftijdscategorie 51-55 jaar.
De luisteraar met email jan.van.vliet@gmail.com     valt in de leeftijdscategorie 56-60 jaar.
ET Listener                                                                    att age_category
MATCH
Predicate: De luisteraar met email <email> valt in de leeftijdscategorie <age_category> jaar.
```

FT12

```
De luisteraar met email marco.engelbart@hotmail.com heeft postcodecijfers 5343.
De luisteraar met email jan.van.vliet@gmail.com     heeft postcodecijfers 1234.
ET Listener                                                               ET PostalCode
MATCH                                                                     ID: att postal_code_numbers
RT PostalCodeOfListener between ET Listener and ET PostalCode
Predicate: De luisteraar met email <email> heeft postcodecijfers <postal_code_numbers>.
```

FT13

```
Op de stemlijst van luisteraar met email marco.engelbart@hotmail.com voor de top 2000 van 2020 staat het nummer Come
Together van The Beatles uit de standaardlijst.
Op de stemlijst van luisteraar met email marco.engelbart@hotmail.com voor de top 2000 van 2020 staat het nummer Purple
Haze van Jimi Hendrix uit de standaardlijst.
ET VoteList                                                                                    ET Song
ID: ET Listener + ET Top2000List                                                               MATCH
    MATCH         MATCH
RT SongOfVote between ET VoteList and ET Song
RT Top2000ListVoteList between ET VoteList (dep) and ET Top2000List
RT VoteListOfListener between ET VoteList (dep) and ET Listener

Predicate: Op de stemlijst van luisteraar met email <email> voor de top 2000 van <edition_year> staat het nummer
<song_name> van <artist_name> uit de standaardlijst.
```

FT14

```
Op de stemlijst van de luisteraar met email marco.engelbart@hotmail.com voor de top 2000 van 2020 staat als vrije keuze
het nummer Dust My Broom van Elmore james.
```

Op de stemlijst van de luisteraar met email marco.engelbart@hotmail.com voor de top 2000 van 2020 staat als vrije keuze het nummer Mystery Train van Elvis Presley.

```
ID: ET VoteList                                                    ET SongNotOnStandardList
    MATCH                                                          ID: att artist_name + att song_name
RT NotStandardSongOfVoteList between ET VoteList and ET SongNotOnStandardList
Predicate Op de stemlijst van de lijsteraar met email <email> voor de top 2000 van <edition_year> staat als vrije keuze
het nummer <song_name> van <artist_name>.
```

FT15

```
Gorillaz is van het type band.
George Michael is van het type persoon.
ET Artist                    att artist_type
MATCH
Predicate: <artist_name> is van het type <artist_type>
```

FT16

```
De band Blur     is opgericht in 1989.
De band Gorillaz is opgericht in 2000.
ET Band                        att founding_year
ST of ET Artist
      MATCH
Predicate: De band <band_name> is opgericht in <founding_year>.
```

FT17

```
Het land met code UK heet Engeland.
Het land met code NL heet Nederland.
ET Country               att country_name
ID: country_code
Predicate: Het land met code <country_code> heet <country_name>
```

FT18

```
De band Blur     is opgericht in UK.
De band Gorillaz is opgericht in UK.
ET Band                        ET Country
MATCH                          MATCH
RT: BandCountry between ET Band and ET Country
Predicate: De band <artist_name> is opgericht in <country_code>.
```

FT19

```
De band The white stripes is opgeheven in 2011.
De band The Beatles       is opgeheven in 1970.
ET Band                                 att dissolution_year
MATCH
Predicate: De band <band_name> is opgeheven in <dissolution_year>.
```

FT20

```
De echte naam van 2pac is Tupac Shakur.
De echte naam van Adele is Adele Adkins.
                  ET Person      att person_name
                  ST of ET Artist
                           MATCH
Predicate: De echte naam van <artist_name> is <person_name>
```

FT20

```
De persoon Damon Albarn is geboren op 23 maart 1968.
De persoon Jack White   is geboren op 9 juli 1975.
ET Person                             att birth_date
MATCH
Predicate: De persoon <artist_name> is geboren op <birth_date>.
```

FT21

```
De persoon Meg white   is overleden in het jaar 2021.
De persoon John Lennon is overleden in het jaar 1980.
ET Person                                  att death_year
MATCH
Predicate: De artiest <artist_name> is overleden in het jaar <death_year>.
```

FT22

```
De persoon Jack White   is/was lid van de band The White Stripes.
De persoon Damon Albarn is/was lid van de band Blur.
ET Person                                          ET Band
MATCH                                              MATCH
RT: BandMembership between ET Artist and ET Band
Predicate: De persoon <artist_name> is/was lid van de band <band_name>.
```

FT23

```
Top 40              is a chart in NL
Singles Chart       is a chart in UK
ET: Chart                        ET Country
ID: chart_name                   MATCH
RT: CountryOfChart between ET Chart and ET Country
Predicate:<chart_name> is a chart in <country_code>.
```

FT24

```
The song Seven Nation Army by The White Stripes reached on the NL Top 40 as its highest position 22.
The song Seven Nation Army by The White Stripes reached on the UK Singles Chart as its highest position 7.
The song Seven Nation Army by The White Stripes reached on the USA Billboard Hot 100 as its highest position 75.
```

```
The song Clint Eastwood by Gorillaz reached on the NL Top 40 as its highest position 27.
The song Clint Eastwood by Gorillaz reached on the UK Singles chart as its highest position 4.
The song Clint Eastwood by Gorillaz reached on the USA Billboard Hot 100 as its highest position 57.
ET: SongOnChart                                                                    att position
ID: ET Song + ET Chart
    MATCH     MATCH
RT: SongOfSongOnChart between ET SongOnChart (dep) and ET Song
RT: ChartOfSongOnChart between ET SongOnChart (dep) and ET Chart
Predicate: <song_name> of <artist_name> reached on the <chart_name> as its highest position <position>.
```

FT25

```
The song Seven Nation Army by The White Stripes stayed on the NL Top 40 for 5 weeks.
The song Seven Nation Army by The White Stripes stayed on the UK Singles Chart for 9 weeks.
The song Seven Nation Army by The White Stripes stayed on the USA Billboard Hot 100 for 1 week.
The song Clint Eastwood by Gorillaz stayed on the NL Top 40 for 6 weeks.
The song Clint Eastwood by Gorillaz stayed on the UK Singles Chart for 17 weeks.
The song Clint Eastwood by Gorillaz stayed on the USA Billboard Hot 100 for 11 weeks.
ET: SongOnChart                                                  att duration_in_weeks
MATCH
Predicate: The song <song_name> of <artist_name> stayed on the <country_code> <chart_name> for < duration_in_weeks>.
```

FT26

```
The song Bohemian Rhapsody by Queen reached on the Top 10 ranking of Arnhem in 2019 position 1.
The song Hotel California by Eagle reached on the Top 10 ranking of Arnhem in 2019 position 2.
The song Piano Man by Billy Joel reached on the Top 10 ranking of Arnhem in 2019 position 3.
ET: SongOnTop10List                                                        att position
ID: ET Song + ET Top10List
    MATCH     ID: ET Municipality + ET Top2000List
                  MATCH             MATCH
RT: SongOfSongOnTop10List between ET SongOnTop10List (dep) and ET Song
RT: Top10ListOfSongOnTop10List between ET SongOnTop10List (dep) and ET Top10List
Predicate: The <song_name> by <artist_name> reached on the Top 10 ranking of <municipality_name> in <edition_year> as its
highest position <position>.
```

FT27

The postcode 6874 belongs to the municipality Renkum.
The postcode 6831 belongs to the municipality Arnhem.
ET: PostalCode ET: Municipality
Match ID: att municipality_code
RT:PostalCodeMunicipality between PostalCode and Municipality.
Predicate:The <postal_code> belongs to the <municipality_code>

FT28

In the postcode 6860 there is a place called Oosterbeek.
In the postcode 6831 there is a place called Arnhem.
ET: PostalCode ET: Place
Match ID: att place_name
RT:PostalCodePlace between PostalCode and Place.
Predicate: In the <postal_code> there is a place called <place_name>.

FT29

Dust in The Wind van Kansas van editie 2022 wordt op 31-12-2022 ergens vanaf 00:00 u afgespeeld.
Ne Me Quitte Pas van Jacques Brel van editie 2022 wordt op 31-12-2022 ergens vanaf 00:00 u afgespeeld.
ET Top2000ListEntry att start_datetime
MATCH
Predicate: <song_name> van <artist_name> van editie <edition_year> wordt op <start_datetime> ergens vanaf <start_datetime> u afgespeeld.

FT30

DJ met code JVI heeft als voornaam Jeroen.
DJ met code RVV heeft als voornaam Rick.
ET: DJ att first_name
ID: att code
Predicate: DJ met code <code> heeft als voornaam <first_name>.

FT31

DJ met code JVI heeft als tussenvoegsel van.
DJ met code JKV heeft als tussenvoegsel in de.
ET: DJ att prefix
MATCH
Predicate: DJ met code <code> heeft als tussenvoegsel <prefix>.

FT32

```
DJ met code JVI heeft als achternaam Inkel.
DJ met code JKV heeft als achternaam Kijk in de Vegt.
ET: DJ                              att last_name
MATCH
Predicate: DJ met code <code> heeft als achternaam <last_name>.
```

FT33

```
DJ met code JVI heeft in 2021 vanaf 00:00 u nummers gedraaid.
DJ met code JKV heeft in 2021 vanaf 08:00 u nummers gedraaid.
DJ met code JWR heeft in 2021 vanaf 08:00 u nummers gedraaid.
ET: DJInYear                        att start_time
ID: ET DJ + ET Top2000List
    MATCH   MATCH
RT: DJOfDJInYear between ET DJInYear (dep) and ET DJ
RT: DJOfTop2000List between ET DJInYear (dep) and ET Top2000List
Predicate: DJ met code <code> heeft in <edition_year> vanaf <start_time> u nummer gedraaid.
```

BIJLAGE D. CLEANUP_QUERIES.SQL

Zie cleanup_queries.sql voor volledige uitwerking met bijbehorende aanmaak van eventuele tabellen en constraints.

```sql
-- [CQSQL1] Haal alle artiesten op die geen duo zijn of band. Maar wel een samenwerkings verband hebben
INSERT INTO Top2000_cleaned.dbo.Artist (artist_name)
       SELECT DISTINCT TRIM(CHAR(160) FROM TRIM(value))  as Artiest
       FROM Top2000.dbo.Song as S
       CROSS APPLY string_split(REPLACE(REPLACE(S.Artiest, '&', '#'), 'ft.', '#'), '#')
       LEFT JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
       WHERE A.artist_name is NULL
GO

-- [CQSQL2] Haal bands/vaste duo's op.
INSERT INTO Top2000_cleaned.dbo.Artist (artist_name)
SELECT DISTINCT S.Artiest
FROM Top2000.dbo.Song as S
INNER JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
GO


-- [SQSQL3] Kopieer alle songs met dat geen band of duo is naar Song en haal de hoofd artiest eruit.
INSERT INTO Top2000_cleaned.dbo.Song (titel, release_year, duration, old_componist, artist)
SELECT S.titel as titel, S.Jaar as release_year, S.Speelduur as duration, S.Componist as old_componist,
       TRIM(CHAR(160) FROM
              CASE
                     WHEN CHARINDEX('ft.', Artiest) > 0
                            THEN TRIM(LEFT(Artiest, CHARINDEX('ft.', Artiest) - 1))
                     WHEN CHARINDEX('&', Artiest) > 0
                            THEN TRIM(LEFT(Artiest, CHARINDEX('&', Artiest) - 1))
                     ELSE TRIM(Artiest)
              END
       ) AS artist
FROM Top2000.dbo.Song as S
LEFT JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
WHERE A.artist_name is NULL
GO

-- [SQSQL4] Kopieer alle songs van vaste duo's of bands
INSERT INTO Top2000_cleaned.dbo.Song (titel, release_year, duration, old_componist, artist)
SELECT S.titel as titel, S.Jaar as release_year, S.Speelduur as duration, S.Componist as old_componist, S.Artiest as artist
FROM Top2000.dbo.Song as S
INNER JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
```

```sql
GO


-- [CQSQL5] Zet alle featured artiesten over
INSERT INTO Top2000_cleaned.dbo.SongFeaturedArtist (song_titel, artist, featured_artist)
SELECT *
FROM (
	SELECT titel as titel,
		TRIM(CHAR(160) FROM
			CASE
				WHEN CHARINDEX('ft.', Artiest) > 0
					THEN TRIM(LEFT(Artiest, CHARINDEX('ft.', Artiest) - 1))
				WHEN CHARINDEX('&', Artiest) > 0
					THEN TRIM(LEFT(Artiest, CHARINDEX('&', Artiest) - 1))
				ELSE TRIM(Artiest)
			END
		) AS artist,
		TRIM(CHAR(160) FROM
			CASE
				WHEN CHARINDEX('ft.', Artiest) > 0
					THEN TRIM(SUBSTRING(Artiest, CHARINDEX('ft.', Artiest) + LEN('ft.'), LEN(Artiest)))
				WHEN CHARINDEX('&', Artiest) > 0
					THEN TRIM(SUBSTRING(Artiest, CHARINDEX('&', Artiest) + LEN('&'), LEN(Artiest)))
				ELSE NULL
			END
		) AS featured_artist
	FROM Top2000.dbo.Song as S
	LEFT JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
	WHERE A.artist_name is NULL
) sub
WHERE featured_artist IS NOT NULL
GO


-- [CQSQL6] Haal alle unieke componisten op uit Top2000
INSERT INTO Top2000_cleaned.dbo.Componist (componist_name)
SELECT DISTINCT TRIM(value) as componist_name
FROM Top2000_cleaned.dbo.Song
CROSS APPLY string_split(old_componist, ',')
GO
```

```sql
-- [CQSQL7] Componisten weer koppelen aan de song waar zij bij horen
INSERT INTO Top2000_cleaned.dbo.SongComponist (titel, artist, componist_name)
SELECT titel, artist, TRIM(s.value) as componist_name
FROM Top2000_cleaned.dbo.Song
CROSS APPLY string_split(old_componist, ',') s
GO


-- [CQSQL8] Insert Alle genres voor losse artiesten en songs met featured artiesten
INSERT INTO Top2000_cleaned.dbo.SongGenre (titel, genre, artist)
SELECT S.titel, SG.genre,
       TRIM(CHAR(160) FROM
              CASE
                     WHEN CHARINDEX('ft.', S.Artiest) > 0
                            THEN TRIM(LEFT(S.Artiest, CHARINDEX('ft.', S.Artiest) - 1))
                     WHEN CHARINDEX('&', S.Artiest) > 0
                            THEN TRIM(LEFT(S.Artiest, CHARINDEX('&', S.Artiest) - 1))
                     ELSE TRIM(S.Artiest)
              END
       ) AS artist
FROM Top2000.dbo.Song as S
INNER JOIN Top2000.dbo.SongGenre as SG ON SG.titel = S.titel AND SG.artiest = S.Artiest
LEFT JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
WHERE A.artist_name is NULL
GO

-- [CQSQL9] Insert alle genres van bands en vaste duo's
INSERT INTO Top2000_cleaned.dbo.SongGenre (titel, genre, artist)
SELECT S.titel, SG.genre, S.Artiest as artist
FROM Top2000.dbo.Song as S
INNER JOIN Top2000.dbo.SongGenre as SG ON SG.titel = S.titel AND SG.artiest = S.Artiest
INNER JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
GO


-- [CQSQL16]
INSERT INTO Top2000_cleaned.dbo.Genre (genre_name)
SELECT DISTINCT genre
FROM Top2000_cleaned.dbo.SongGenre
```

```
GO


-- [CQSQL17]
INSERT INTO Top2000_cleaned.dbo.Top2000Lijst (edition_year, position, titel, artist)
SELECT editiejaar as edition_year, positie as position, L.Titel as titel,
       TRIM(CHAR(160) FROM
              CASE
                     WHEN CHARINDEX('ft.', S.Artiest) > 0
                            THEN TRIM(LEFT(S.Artiest, CHARINDEX('ft.', S.Artiest) - 1))
                     WHEN CHARINDEX('&', S.Artiest) > 0
                            THEN TRIM(LEFT(S.Artiest, CHARINDEX('&', S.Artiest) - 1))
                     ELSE TRIM(S.Artiest)
              END
       ) AS artist
FROM Top2000.dbo.Top2000Lijst as L
INNER JOIN Top2000.dbo.Song as S ON L.Artiest = S.Artiest and L.Titel = S.titel
LEFT JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
WHERE A.artist_name is NULL
GO

-- [CQSQL18]
INSERT INTO Top2000_cleaned.dbo.Top2000Lijst (edition_year, position, titel, artist)
SELECT editiejaar as edition_year, positie as position, L.Titel as titel, S.artiest
FROM Top2000.dbo.Top2000Lijst as L
INNER JOIN Top2000.dbo.Song as S ON L.Artiest = S.Artiest and L.Titel = S.titel
INNER JOIN Top2000_cleaned.dbo.ampersand_artists as A ON S.Artiest = A.artist_name
GO
```

BIJLAGE E.

```
-- [ESQL1]
SELECT
       Top2017.POSITION AS NR,
       (Top2016.POSITION - Top2017.POSITION) AS VERLOOP,
       Top2017.SONG_NAME AS TITEL,
       Top2017.ARTIST_NAME AS ARTIEST,
       SongInfo.RELEASE_YEAR AS JAAR
FROM
```

```
	[Top2000_2.0].[dbo].[TOP2000LISTENTRY] Top2017
LEFT JOIN
	[Top2000_2.0].[dbo].[TOP2000LISTENTRY] Top2016
	ON Top2017.SONG_NAME = Top2016.SONG_NAME
	AND Top2017.ARTIST_NAME = Top2016.ARTIST_NAME
	AND Top2016.EDITION_YEAR = 2016
LEFT JOIN
	[Top2000_2.0].[dbo].[SONG] SongInfo
	ON Top2017.SONG_NAME = SongInfo.SONG_NAME
	AND Top2017.ARTIST_NAME = SongInfo.ARTIST_NAME
WHERE
	Top2017.EDITION_YEAR = 2017
ORDER BY
Top2017.POSITION;

-- [ESQL2]
SELECT TOP 10 ARTIST_NAME, SONG_NAME, COUNT(*) as aantalstemmen, ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER (), 2) AS
percentage_van_totaal
FROM SONGOFVOTE as SV
JOIN (
	SELECT V.*
	FROM LISTENER AS L
	JOIN POSTALCODE AS P on L.POSTAL_CODE_NUMBERS = P.POSTAL_CODE_NUMBERS
	JOIN VOTELIST AS V on L.EMAIL = V.EMAIL
	WHERE P.MUNICIPALITY_CODE = 150 AND V.EDITION_YEAR = 2017
) sub
ON sub.EMAIL = SV.EMAIL and sub.EDITION_YEAR = SV.EDITION_YEAR
GROUP BY ARTIST_NAME, SONG_NAME
ORDER BY COUNT(*) DESC
```

BIJLAGE F.

Zie migratie_scriptjes.sql in het ZIP bestand.

OPEN UP
NEW HAN_UNIVERSITY OF APPLIED SCIENCES
HORIZONS.