

Examen (HMIN231)

YANG MINYOUNG
STEFANOVSKI STEFAN

Avril 2020

1 Introduction, Question 1

Nous avons utilisé l'outil Protégé afin de construire notre ontologie. Notre sujet consiste au domaine de la musique (Chanteur, Chanson, Genre et Maison de production).

Nous présentons en dessous nos concepts, rôles ainsi que les explications.

1.1 Concepts simples

- Genre (*Genres de la musique*)
- Song (*Titres des chansons*)
- Person (*Ensemble de personnes*)
- Production company (*Maisons de production*)

1.2 Concepts construits

- $\text{Singer} \sqsubseteq \text{Person} \sqcap \exists \text{Sings} . \text{Song} \sqcap \exists \text{IsSingerOf} . \text{Genre}$
(*Un chanteur est une personne qui chante une/des chanson(s) et d'un/des genre(s) musical/caux*)
- $\text{Producer} \sqsubseteq \text{Person} \sqcap \exists \text{Produced} . \text{Song}$
(*Un producteur est une personne qui produit une/des chanson(s)*)
- $\text{Singer} \sqcap \exists \text{Sings} . \text{Song} \sqcap \exists \text{IsSingerOf} . \text{Genre} \sqsubseteq \text{T}$
(*Un chanteur qui chante une/des chanson(s) et d'un/des genre(s) musical/caux est toujours vrai*)
- $\text{Producer} \sqcap \exists \text{Produced} . \text{Song} \sqsubseteq \text{T}$
(*Un producteur qui produit une/des chansons est toujours vrai*)
- $\text{ProductionCompany} \sqcap \exists \text{ProducedPC} . \text{Song} \sqsubseteq \text{T}$
(*Une maison de production a produit au minimum une chanson*)
- $\text{Song} \sqcap \exists \text{SingBy} . \text{Singer} \sqcap \exists \text{HasSongAs} . \text{Genre} \sqcap \exists \text{ProducedBy} . \text{Producer} \sqcap \exists \text{ProducedIn} . \text{ProductionCompany} \sqsubseteq \text{T}$
(*Une chanson est toujours chantée par un/plusieurs chanteurs, produit par un/plusieurs producteurs et produit dans une maison de production*)

- $\text{Person} \sqcap \exists \text{Sings} . \text{Song} \sqcap \exists \text{Produced} . \text{Song} \sqsubseteq \text{Singer} \sqcap \text{Producer}$
(Une personne pourrait tre un chanteur et également un producteur si il chante et a produit une/des chanson(s))

1.3 Rôles

- $\text{Sings}(\text{Singer}, \text{Song})$ *(Un chanteur chante une chanson)*
- $\text{Singby}(\text{Song}, \text{Singer})$ *(Une chanson est chantée par un chanteur)*
- $\text{IsGenreOf}(\text{Song}, \text{Genre})$ *(Une chanson est de Genre musicale)*
- $\text{HasSongAs}(\text{Genre}, \text{Song})$ *(Un genre contient comme chanson)*
- $\text{Produced}(\text{Producer}, \text{Song})$ *(Un producteur a produit une chanson)*
- $\text{IsProducedBy}(\text{Song}, \text{Producer})$ *(Une chanson a été produit par un producteur)*
- $\text{IsSingerOf}(\text{Singer}, \text{Genre})$ *(Un chanteur chante tel genre de musique)*
- $\text{HasSingerAs}(\text{Genre}, \text{Singer})$ *(Un genre de musique contient tel chanteur)*
- $\text{ProducedPC}(\text{Production Company}, \text{Song})$ *(Une maison de production a produit telle chanson)*
- $\text{ProducedIn}(\text{Song}, \text{Production Company})$ *(Une chanson a été produit dans telle maison de production)*

2 Question 2

En *ALCNR*, c'est possible de définir les chansons chantées par plusieurs chanteurs, par exemple, une chanson qui est chantée par minimum 2 chanteurs (duo) peut se traduire comme : $\geq 2 \text{ SingBy}$. Par contre, si nous souhaitons ajouter une information telle que ces deux chanteurs sont également les producteurs de la chanson, ce n'est plus possible avec *ALCNR*.

Dans ce cas, nous pouvons utiliser l'extension "Qualified number restriction : *Q*" donc *ALCNQ* car elle permet de déduire une information de plus à partir du rôle donné. Cela se traduira comme : $\geq 2 \text{ SingBy.Producer}$

3 Question 3

- $\text{Singer}(\text{Aloe Blacc})$
- $\text{Producer}(\text{Kiril Dzaikovski})$
- $\text{Production Company}(\text{Sony Music Entertainment})$
- $\text{Sings}(\text{Shakira}, \text{Waka Waka})$
- $\text{Singby}(\text{Get lucky}, \text{Daft Punk})$
- $\text{Singby}(\text{Get lucky}, \text{Pharell Williams})$
- $\text{IsGenreOf}(\text{Reminder}, \text{Electro})$
- $\text{HasSongAs}(\text{Pop}, \text{I need a dollar})$
- $\text{Produced}(\text{John Hill}, \text{Waka Waka})$
- $\text{IsProducedBy}(\text{Waka Waka}, \text{John Hill})$
- $\text{IsSingerOf}(\text{Shakira}, \text{Pop})$

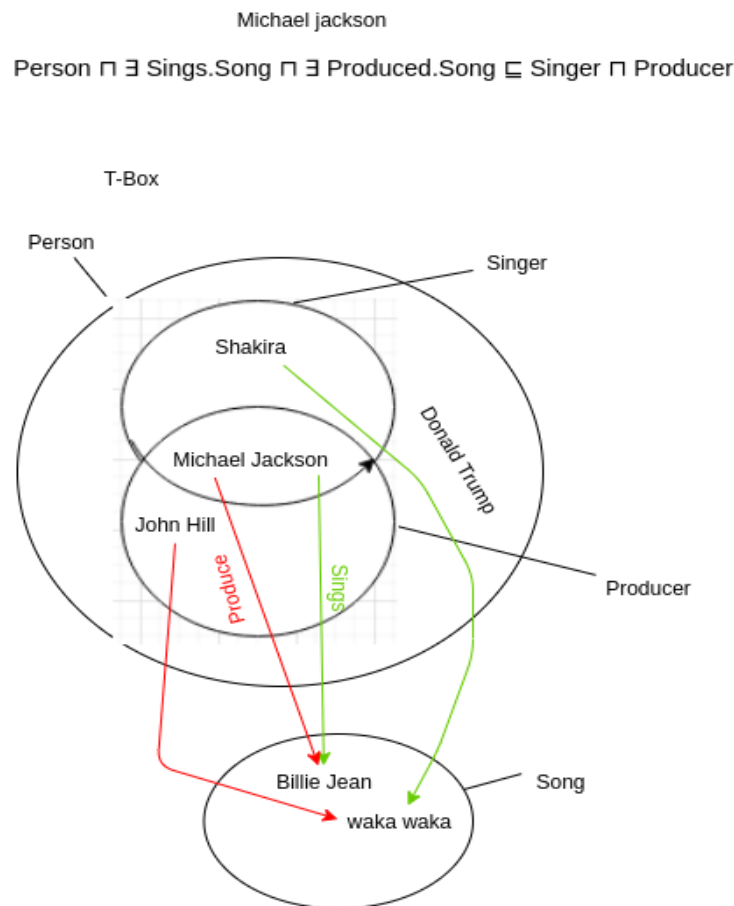
- hasSingerAs(Pop,Shakira)
- ProducedPC(Daft Life Columbia, Get lucky)
- ProducedIn(Diamonds,Sony Music Entertainment)

4 Question 4

Pour cette question, nous avons pris comme exemple *Michael Jackson* qui n'est pas dans notre A-Box.

A partir de notre base de connaissances, nous pouvons déduire que *Michael Jackson*, qui est dans le concept *Person*, est en même temps un chanteur et un producteur.

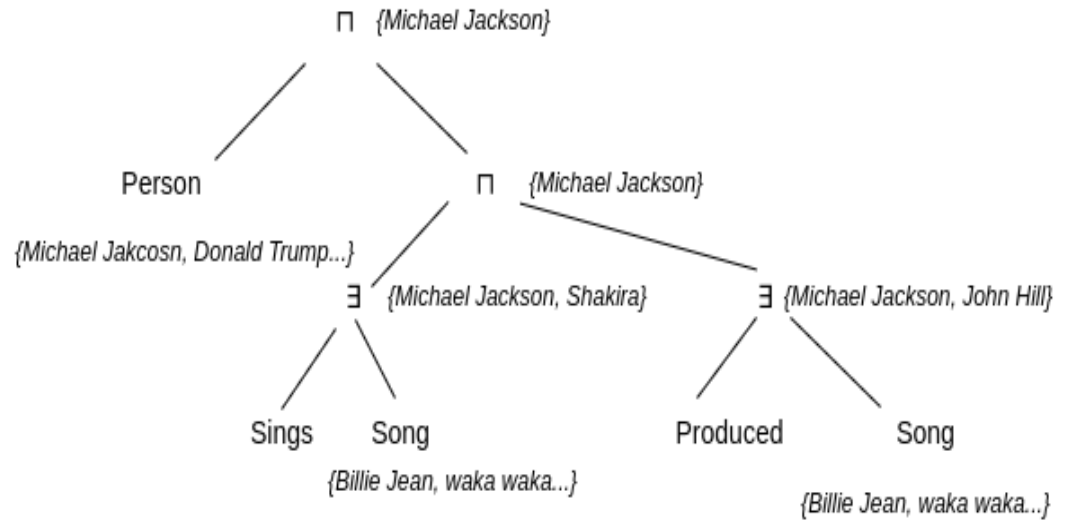
Voici notre raisonnement :



Ce schéma représente notre T-Box avec quelques instances de personne et de chanson. Nous avons des relations *Sings* et *Produced* qui partent des personnes

vers les chansons.

Notre exemple *Michael Jackson* est une personne, il a chanté et produit la chanson *Billie Jean*.



Nous avons défini précédemment (le dernier concept construit) que si une personne chante ET a produit une/des chanson(s), alors cette personne est en même temps un chanteur et un producteur. A partir de cette base, nous avons pu construire cet arbre de l'interprétation et conclure que *Michael Jackson*, qui est bien à la fois un chanteur et un producteur.

5 Question 5

Singer \sqcap Producer \sqsubseteq Person $\sqcap \exists \text{Sings.Song} \sqcap \exists \text{Produced.Song}$?

$\text{Singer} \sqcap \text{Producer} \sqcap \neg(\text{Person} \sqcap \exists \text{Sings.Song} \sqcap \exists \text{Produced.Song}) : x$

$\text{Singer} \sqcap \text{Producer} \sqcap (\neg \text{Person} \sqcup \forall \text{Sings.} \neg \text{Song} \sqcup \forall \text{Produced.} \neg \text{Song}) : x$

$\text{Person} \sqcap \exists \text{Sings.Song} \sqcap \text{Person} \sqcap \exists \text{Produced.Song} \sqcap (\neg \text{Person} \sqcup \forall \text{Sings.} \neg \text{Song} \sqcup \forall \text{Produced.} \neg \text{Song}) : x$

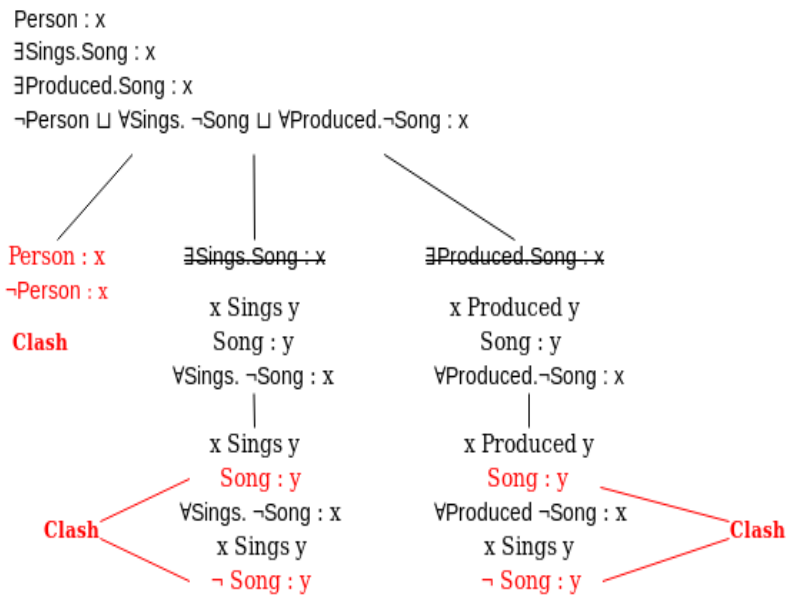


Figure 1

Les branches sont toutes fermées (elles clashent tous), donc la formule est insatisfiable, le modèle est valide. D'où il existe la conséquence sémantique : $\text{Singer} \sqcap \text{Producer} \models \text{Person} \sqcap \exists \text{Sings.Song} \sqcap \exists \text{Produced.Song}$

6 Question 6

$\text{Person} \sqsubseteq \text{Singer} \sqcup \text{Producer}$

$\text{Person} \sqcap \neg (\text{Singer} \sqcup \text{Producer}) \sqsubseteq \perp$

$\text{Person} \sqcap \neg \text{Singer} \sqcap \neg \text{Producer} : x$

$\text{Person} \sqcap \neg (\text{Person} \sqcap \exists \text{Sings.Song}) \sqcap \neg (\text{Person} \sqcap \exists \text{Produced.Song}) : x$

$\text{Person} \sqcap (\neg \text{Person} \sqcup \neg \exists \text{Sings.Song}) \sqcap (\neg \text{Person} \sqcup \neg \exists \text{Produced.Song}) : x$

$\text{Person} \sqcap (\neg \text{Person} \sqcup \forall \text{Sings.}\neg \text{Song}) \sqcap (\neg \text{Person} \sqcup \forall \text{Produced.}\neg \text{Song}) : x$

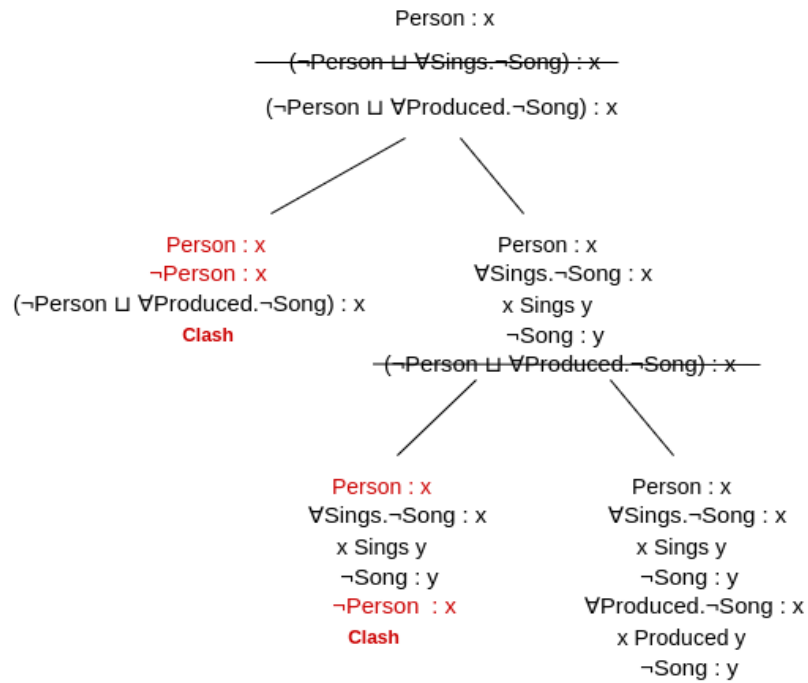


Figure2

L'inclusion de la Figure2 n'est pas déductible de notre T-Box, car il existe une branche où nous pouvons développer jusqu'à l'infinité à cause des deux rôles $\forall \text{Sings.}\neg \text{Song}$ et $\forall \text{Produced.}\neg \text{Song}$

7 Question 7

Voir le fichier RC_Musique.owl.

Classes

Genre

Person

Producer

$\text{Producer} \sqsubseteq \text{Person}$
 $\text{Producer} \sqsubseteq \text{Person} \sqcap \exists \text{Produced Song}$

Production_Company

Singer

$\text{Singer} \sqsubseteq \text{Person} \sqcap \exists \text{Sings Song}$
 $\text{Singer} \sqsubseteq \text{Person}$

Song

Object properties

HasSongAs

$\text{HasSongAs} \equiv \text{IsGenreOf}^-$
 $\exists \text{HasSongAs Thing} \sqsubseteq \text{Genre}$
 $\top \sqsubseteq \forall \text{HasSongAs Song}$

IsGenreOf

$\text{HasSongAs} \equiv \text{IsGenreOf}^-$
 $\exists \text{IsGenreOf Thing} \sqsubseteq \text{Song}$
 $\top \sqsubseteq \forall \text{IsGenreOf Genre}$

IsSingerOf

$\text{IsSingerOf} \equiv \text{hasSingerAs}^-$
 $\exists \text{IsSingerOf Thing} \sqsubseteq \text{Singer}$
 $\top \sqsubseteq \forall \text{IsSingerOf Genre}$

Produced

$\text{Produced} \equiv \text{ProducedBy}^-$
 $\exists \text{Produced Thing} \sqsubseteq \text{Producer}$
 $\top \sqsubseteq \forall \text{Produced Song}$

ProducedBy

$\text{Produced} \equiv \text{ProducedBy}^-$
 $\exists \text{ ProducedBy Thing} \sqsubseteq \text{Song}$
 $\top \sqsubseteq \forall \text{ ProducedBy Producer}$

ProducedIn

$\sqsubseteq \text{topObjectProperty}$
 $\text{ProducedIn} \equiv \text{ProducedPC}^-$
 $\exists \text{ ProducedIn Thing} \sqsubseteq \text{Song}$
 $\top \sqsubseteq \forall \text{ ProducedIn Production_Company}$

ProducedPC

$\sqsubseteq \text{topObjectProperty}$
 $\text{ProducedIn} \equiv \text{ProducedPC}^-$
 $\exists \text{ ProducedPC Thing} \sqsubseteq \text{Production_Company}$
 $\top \sqsubseteq \forall \text{ ProducedPC Song}$

SingBy

$\text{SingBy} \equiv \text{Sings}^-$
 $\exists \text{ SingBy Thing} \sqsubseteq \text{Song}$
 $\top \sqsubseteq \forall \text{ SingBy Singer}$

Sings

$\text{SingBy} \equiv \text{Sings}^-$
 $\exists \text{ Sings Thing} \sqsubseteq \text{Singer}$
 $\top \sqsubseteq \forall \text{ Sings Song}$

hasSingerAs

$\text{IsSingerOf} \equiv \text{hasSingerAs}^-$
 $\exists \text{ hasSingerAs Thing} \sqsubseteq \text{Genre}$
 $\top \sqsubseteq \forall \text{ hasSingerAs Singer}$

Individuals

Aloe_Blacc

$\text{Aloe_Blacc} : \text{Singer}$
 $\text{Sings}(\text{Aloe_Blacc}, \text{I_need_a_dollar})$
 $\text{IsSingerOf}(\text{Aloe_Blacc}, \text{Pop})$

Bright_lights

Bright_lights : Song
ProducedBy(Bright_lights, Gary_Clark)
SingBy(Bright_lights, Gary_Clark)
IsGenreOf(Bright_lights, Rock)

Cant_remember_to_forget_you

Cant_remember_to_forget_you : Song
ProducedIn(Cant_remember_to_forget_you, EMI_Capitol_Toshiba)
SingBy(Cant_remember_to_forget_you, Shakira)
ProducedBy(Cant_remember_to_forget_you, John_Hill)
SingBy(Cant_remember_to_forget_you, Rihanna)
IsGenreOf(Cant_remember_to_forget_you, Pop)

Country

Country : Genre
HasSongAs(Country, how_you_remind_me)
hasSingerAs(Country, Nickelback)

Daft_Life_Columbia

Daft_Life_Columbia : Production_Company
ProducedPC(Daft_Life_Columbia, Get_lucky)

Daft_Punk

Daft_Punk : Singer
Daft_Punk : Person
Daft_Punk : Producer
Produced(Daft_Punk, Get_lucky)
IsSingerOf(Daft_Punk, Electro)
Sings(Daft_Punk, Get_lucky)
IsSingerOf(Daft_Punk, Pop)
IsSingerOf(Daft_Punk, Funk)

Diamonds

Diamonds : Song
ProducedIn(Diamonds, Sony_Music_Entertainment)
SingBy(Diamonds, Rihanna)
IsGenreOf(Diamonds, Pop)

EMI_Capitol_Toshiba

EMI_Capitol_Toshiba : Production_Company
ProducedPC(EMI_Capitol_Toshiba, I_want_to_break_free)

Electro

Electro : Genre
hasSingerAs(Electro, Daft_Punk)
HasSongAs(Electro, Reminder)
hasSingerAs(Electro, Kiril_Dzajkovski)

Funk

Funk : Genre
HasSongAs(Funk, Get_lucky)
hasSingerAs(Funk, Daft_Punk)

Gary_Clark

Gary_Clark : Singer
IsSingerOf(Gary_Clark, Rock)
Sings(Gary_Clark, Bright_lights)

Get_lucky

Get_lucky : Song
SingBy(Get_lucky, Daft_Punk)
IsGenreOf(Get_lucky, Funk)
SingBy(Get_lucky, Pharell_Williams)
ProducedBy(Get_lucky, Daft_Punk)
IsGenreOf(Get_lucky, Pop)
ProducedIn(Get_lucky, Daft_Life_Columbia)

I_need_a_dollar

I_need_a_dollar : Song
IsGenreOf(I_need_a_dollar, Pop)
SingBy(I_need_a_dollar, Aloe_Blacc)

I_want_to_break_free

I_want_to_break_free : Song
IsGenreOf(I_want_to_break_free, Rock)
ProducedIn(I_want_to_break_free, EMI_Capitol_Toshiba)
ProducedBy(I_want_to_break_free, Reinhold_Mack)
SingBy(I_want_to_break_free, Queen)
ProducedBy(I_want_to_break_free, Queen)

John_Hill

John_Hill : Producer
Produced(John_Hill, waka_waka)
Produced(John_Hill, Cant_remember_to_forget_you)

Kiril_Dzajkovski

Kiril_Dzajkovski : Singer
Kiril_Dzajkovski : Person
Kiril_Dzajkovski : Producer
IsSingerOf(Kiril_Dzajkovski, Electro)
Sings(Kiril_Dzajkovski, Reminder)
Produced(Kiril_Dzajkovski, Reminder)

Nickelback

Nickelback : Singer
IsSingerOf(Nickelback, Country)
Sings(Nickelback, how_you_remind_me)
IsSingerOf(Nickelback, Rock)

Pharell_Williams

Pharell_Williams : Producer
Pharell_Williams : Singer
Pharell_Williams : Person
IsSingerOf(Pharell_Williams, Pop)
Sings(Pharell_Williams, Get_lucky)

Pop

Pop : Genre
hasSingerAs(Pop, Aloe_Blacc)
HasSongAs(Pop, Diamonds)
HasSongAs(Pop, I_need_a_dollar)
HasSongAs(Pop, Cant_remember_to_forget_you)
hasSingerAs(Pop, Pharell_Williams)
hasSingerAs(Pop, Rihanna)
hasSingerAs(Pop, Shakira)
HasSongAs(Pop, Get_lucky)
HasSongAs(Pop, waka_waka)

Queen

Queen : Singer
IsSingerOf(Queen, Rock)
Sings(Queen, I_want_to_break_free)

Reinhold_Mack

Reinhold_Mack : Producer
Produced(Reinhold_Mack, I_want_to_break_free)

Reminder

Reminder : Song
IsGenreOf(Reminder, Electro)
SingBy(Reminder, Kiril_Dzajkovski)
ProducedBy(Reminder, Kiril_Dzajkovski)

Rihanna

Rihanna : Singer
Sings(Rihanna, Diamonds)
Sings(Rihanna, Cant_remember_to_forget_you)
IsSingerOf(Rihanna, Pop)

Rock

Rock : Genre
HasSongAs(Rock, I_want_to_break_free)
HasSongAs(Rock, Bright_lights)
hasSingerAs(Rock, Queen)
hasSingerAs(Rock, Gary_Clark)
HasSongAs(Rock, how_you_remind_me)
hasSingerAs(Rock, Nickelback)

Shakira

Shakira : Singer
Sings(Shakira, waka_waka)
Sings(Shakira, Cant_remember_to_forget_you)
IsSingerOf(Shakira, Pop)

Sony_Music_Entertainment

Sony_Music_Entertainment : Production_Company
ProducedPC(Sony_Music_Entertainment, Cant_remember_to_forget_you)
ProducedPC(Sony_Music_Entertainment, waka_waka)
ProducedPC(Sony_Music_Entertainment, Diamonds)

how_you_remind_me

how_you_remind_me : Song
IsGenreOf(how_you_remind_me, Country)
SingBy(how_you_remind_me, Nickelback)
IsGenreOf(how_you_remind_me, Rock)

waka_waka

waka_waka : Song
ProducedIn(waka_waka, EMI_Capitol_Toshiba)
IsGenreOf(waka_waka, Pop)
SingBy(waka_waka, Shakira)
ProducedBy(waka_waka, John_Hill)