**Rapport Projet Bases de Données Déductives**



*Master 1 informatique  
Université Claude Bernard Lyon 1*

*Année 2022 - 2023*

Étudiants Encadrant

ENG Charles p1403762 BONIFATI Angela  
TANG Kévin p1501263

# 1 Implémentation

Notre programme est composé de plusieurs parties.

La première partie intitulée “Séparation des EDB et IDB”, comme son nom l’indique, le programme lit le fichier en entrée et sépare chaque ligne pour les identifier comme EDB ou IDB.

La deuxième partie est le traitement des EDB. Le but est de pouvoir facilement exploiter ces données. Pour cela, nous avons fait en sorte que chaque ligne de l’EDB devienne une ligne d’un tableau regroupant tous les EDB.

Par exemple :

artists(123,'KeanuReeves','americaine').

artists(415,'MarionCotillard','francaise').

roles(1,415,'Stephanie',21000).

seront rangés de la façon suivante :

[['artists', 123, 'KeanuReeves', 'americaine'],

['artists', 415, 'MarionCotillard', 'francaise'],

['roles', 1, 415, 'Stephanie', 21000]]

Une troisième partie regroupe toutes les fonctions utiles au traitement des IDB. Il y a par exemple des fonctions pour séparer une règle selon un caractère spécifique, une fonction pour fusionner les résultats des différentes règles ou des fonctions pour gérer les fonctions d’agrégation, etc…

La dernière partie contient la fonction principale de notre programme, il s’agit de notre interprétation de la boucle écrite dans le sujet du projet. Elle réalise la grande majorité des actions du programme et affiche les résultats dans un fichier *output.txt*.

# 2 Evaluation

1. Entrée sans fonction d'agrégation :

**r(B,E) :- artists(A,B,\_), films(\_,E,\_,\_,\_), roles(\_,A,\_,\_).**

Le programme reconnaît qu’il n’y a pas de fonction d’agrégation dans la partie head et renvoie donc la phrase “Rule not an aggregation”.

1. Utilisation simple de COUNT sur une table *artists* avec une seule variable :

**COUNT(X) :- artists(X,\_,\_).**

Le programme cherche toutes les valeurs possibles de X dans les EDB et compte le nombre de résultats.

Résultat : 12

1. Utilisation simple de SUM sur une table *roles* avec une seule variable :

**SUM(X) :- roles(\_,\_,\_,X).**

Le programme cherche toutes les valeurs possibles de X dans les EDB et calcule la somme de ces valeurs.

Résultat : 588500

1. Utilisation simple d' AVG sur une table *roles* avec une seule variable :

**AVG(X) :- roles(\_,\_,\_,X).**

Le programme cherche toutes les valeurs possibles de X dans les EDB et calcule la moyenne de ces valeurs.

Résultat : 49041.666666666664

1. Utilisation d’une condition de filtrage :

**COUNT(X) :- artists(X,\_,C), (C='italienne').**

Le programme calcule les résultats possibles pour *artists(X,\_,C)* puis filtre les résultats en fonction de la condition.

Résultat : 2

1. Utilisation simple d’AVG sur une table *films* avec une seule variable :

**AVG(A) :- films(\_,\_,\_,A,\_).**

Résultat : 1996.3333333333333

1. Utilisation de deux règles et d’une condition de filtrage :

**COUNT(X) :- roles(X,A,\_), artists(A,N,\_), (N='KeanuReeves').**

Le programme calcule le résultat de *roles(X,A,\_)* puis de *artists(A,N,\_)* et fusionne leur résultat pour ensuite les filtrer avec la dernière condition.

Résultat : 2

1. Utilisation équivalente à la précédente mais avec une condition de filtrage différente

**COUNT(A) :- artists(A,\_,\_), roles(\_,A,\_,C), (C>50000).**

Résultat : 4

1. Utilisation de deux conditions de filtrage :

**COUNT(X) :- films(X,\_,\_,A,N), (A>2000), (N='anglaise').**

Le programme calcule les valeurs possibles de *films(X,\_,\_,A,N)* puis filtre avec la première condition *(A>2000)* puis avec le deuxième filtre (N='anglaise').

Résultat : 2

1. Utilisation de trois règles et d’une condition de filtrage :

**COUNT(A) :- artists(A,\_,\_), roles(F,A,\_,\_), films(F,\_,\_,Y,\_), (Y<2000).**

Le programme calcule les valeurs possibles de *artists(A,\_,\_)* puis *roles(F,A,\_,\_)* et les fusionne. Il calcule ensuite *films(F,\_,\_,Y,\_)* et fusionne avec les résultats précédents. Enfin, il filtre avec la dernière condition.

Résultat : 6

Nous n’avons pour l’instant pas trouvé de cas où notre programme ne fonctionnerait pas pour résoudre des requêtes contenant des fonctions d’agrégation, mais il serait peut-être possible d’en trouver avec des requêtes plus complexes.