

Université de Mons
Faculté Des Sciences

Projet de bases de données I

Rapport

Professeur :
Jef WIJSEN
Assistant :
Sébastien BONTE

Auteurs :
Julien LADEUZE
François VION



Année académique 2022-2023

1 Introduction

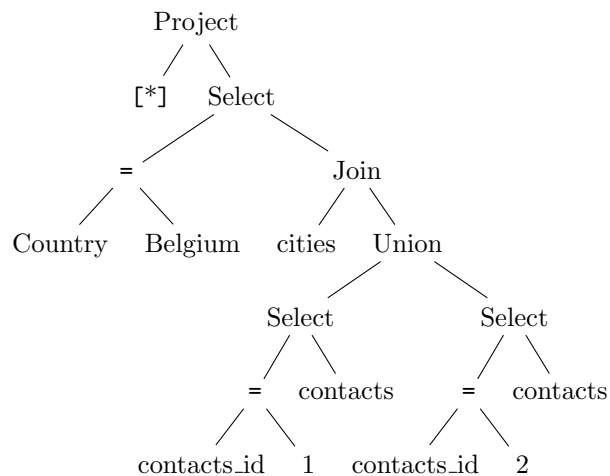
Dans le cadre de notre cours de base de données. Il nous est demandé de réaliser un programme en python qui réalise une conversion d'une requête SPJRUD en requête SQL et de montrer le résultat de cette requête en l'exécutant sur une base de données donnée en paramètre.

2 Choix d'implémentation

2.1 Structure de donnée

Afin de décomposer une requête SPJRUD et de la traduire ensuite en une requête SQL, il a été décidé d'utiliser un arbre syntaxique (SyntaxTree dans le code). Celui implémenté dans le code est légèrement différent de celui montré dans l'énoncé. En effet, il se comporte comme un arbre binaire. Les feuilles sont des attributs, constantes ou tables et les noeuds internes sont des opérations ou des opérandes. Voici-ci dessous un exemple de décomposition de requête avec cet arbre.

```
Project([*],Select(Country,=,Belgium,Join(cities,
Union(Select(contact_id,=,1,contacts),Select(contact_id,=,2,contacts))))))
```



Une fois le SyntaxTree créé, la requête SQL peut être créée en effectuant la traduction de manière postordre (évaluer les fils avant d'évaluer le noeud actuel).

3 Orienté objet

Nous avons décidé de faire sur une grande partie de l'application en orienté objet car ce paradigme permet une implémentation plus simple de certaines fonctions comme par exemples les modules **syntaxTree** et **entities**.

4 Fonctionnalités supplémentaires

4.1 Command Line Interface

Pour éviter d'exécuter le programme à chaque requête, nous avons ajouté un prompt qui permet de faire des requêtes à la suite. Il sera accessible dès que le programme sera lancé. Pour le quitter, il suffira de taper **exit** pour quitter le prompt et terminer l'exécution du programme.

4.1.1 Affichage de l'arbre syntaxique dans le terminal

Afin de débbugger plus facilement le programme, une fonction `display()` a été ajoutée au `SyntaxTree` afin d'afficher celui-ci dans le terminal. Cette fonction a été gardée et est utilisable en entrant **displayTree** dans le terminal après avoir exécuté une requête valide. A noter que cette fonction est une adaptation d'un code trouvé sur internet ¹.

4.1.2 Affichage les différentes tables de la base de données

Pour faciliter les requêtes de l'utilisateur, le programme lui permet d'afficher toutes les tables présentes dans la base de données. Ceci se fait avec la commande **ls table**.

4.1.3 Affichage de tous les attributs d'une relation (*)

Pour projeter tout les attributs d'une relation, le sucre syntaxique **Project([*], nomderelation)** a été implémenté.

4.2 Affichage des requêtes des sessions passées (&)

L'utilisateur a la possibilité de pouvoir charger à nouveau les requêtes syntaxiquement valides des anciennes sessions du programme. Pour cela, il suffit de taper **&** dans le prompt. Ces requêtes sont issues du fichier **request.log**. A noter que si l'utilisateur a atteint la fin des requêtes sauvegardées, alors il recommencera depuis la dernière effectuée. Le module **readline**² a été utilisé pour modifier directement l'entrée utilisateur.

4.3 Manuel d'utilisation

Une documentation a été créée dans le programme, elle est accessible en faisant **python3 main.py -h**. Celle-ci cite les différentes commandes et leur fonctionnement.

1. Voir Stackoverflow

2. Voir Stackoverflow

5 Mode d'emploi

5.1 Executer le programme

Pour exécuter le programme et accéder au prompt, il suffit de taper **python3 main.py nomDeLaBaseDeDonnée.db**. Pour afficher la documentation, **python3 main.py -h**.

5.2 Syntaxe des requêtes SPJRUD

S : **Select**(attribut,=,constante,relation) ou
 Select(attribut,! =,constante,relation)
P : **Project**([attribut1,attribu2,...],relation) ou **Project**([*],relation)
J : **Join**(relation1,relation2)
R : **mRename**(attribut,constante,relation)
U : **Union**(relation1,relation2)
D : **Difference**(relation1,relation2)

5.2.1 Exemples

Une base de données **test.db** est mise à disposition pour effectuer les exemples.

Select(Name,=,Bergen,cities)
Project([*],contacts)
Join(cities,contacts)
Rename(contact_id,id,contacts)
Union(contacts,contacts)
Difference(**Project**([Name],cities),**Project**([Name],cities))