

INFOH303 - Projet de Système d'Inventaire pour un RPG - Partie 2 - Rapport

Prénom	Nom	Matricule
Bilal	Vandenberghe	000593438
Lucas	Verbeiren	000591223
Ethan	Van Ruyskensvelde	000589640
Rares	Radu-Loghin	000590079

1 Méthode d'extraction des données

Pour remplir la base de données, nous avons développé un script `insert.py` (situé dans `src/database/insert.py`). Ce script commence par créer les tables si elles ne sont pas encore présentes dans la base de données (en appelant la fonction `create_database_and_tables()` située dans le fichier `src/database/create.py`). Ensuite, pour chaque fichier présent dans le dossier `data`, nous lisons et extrayons les données, puis tentons de les insérer dans la base de données. Avant chaque insertion, nous effectuons une vérification afin de nous assurer que les données soient valides, comme le type (par exemple `int`, `string`, etc.) et le contenu, en vérifiant notamment si une donnée similaire a déjà été insérée.

Si une donnée est incorrecte (par exemple, si une information est manquante ou si une chaîne de caractères est présente là où un entier est attendu), elle n'est pas ajoutée à la base de données. Ce processus est répété pour chaque donnée contenue dans les fichiers.

Par souci de clarté et de modularité, le traitement de chaque fichier est organisé en fonctions distinctes.

2 Requêtes demandées

Pour exécuter et afficher les différentes requêtes demandées, il suffit de lancer le script suivant :

```
python3 src/database/additional_request.py
```

2.1 Les 10 joueurs ayant le plus d'or

2.1.1 SQL

```
SELECT p.Name, p.Money
FROM Players p
ORDER BY p.Money DESC LIMIT 10;
```

2.1.2 Algèbre relationnelle

Cette requête ne peut pas être exprimée en algèbre relationnelle, car elle suppose une notion d'ordre décroissant sur la quantité d'or et une limitation à 10 résultats. Or, l'algèbre relationnelle ne possède pas d'opérateur de tri ou de limitation. Les résultats sont des ensembles non ordonnés.

2.1.3 Calcul tuple

Cette requête ne peut pas être exprimée en calcul relationnel tuple, car ce formalisme est basé sur la théorie des ensembles et n'inclut ni ordre, ni limite, ni classement.

2.2 Le joueur ayant le plus de personnages de la même classe

2.2.1 SQL

```
SELECT p.Name, c.Class, COUNT(*) AS nbCharacters
FROM Players p
JOIN Characters c ON p.Name = c.Username
GROUP BY p.Name, c.Class
ORDER BY nbCharacters DESC LIMIT 1;
```

2.2.2 Algèbre relationnelle

L'algèbre relationnelle ne contient aucun opérateur d'agrégation ni de comparaison, ce qui rend cette requête impossible dans ce formalisme.

2.2.3 Calcul tuple

Cette requête ne peut pas être exprimée en calcul relationnel tuple, car ce formalisme est basé sur la théorie des ensembles et n'inclut ni ordre, ni limite.

2.3 La quête ayant la plus grosse récompense en or par niveau de difficulté

2.3.1 SQL

```
WITH QuestGold AS (
SELECT q.Name AS QuestName, q.Difficulty, SUM(i.Price * r.Quantity) AS TotalGold
FROM Quests q
JOIN Rewards r ON q.Name = r.QuestName
JOIN Items i ON r.ItemName = i.Name
GROUP BY q.Name, q.Difficulty
),
RankedQuests AS (
SELECT *, ROW_NUMBER() OVER (PARTITION BY Difficulty ORDER BY TotalGold DESC) AS 'rank'
FROM QuestGold
)
SELECT QuestName, Difficulty, TotalGold
FROM RankedQuests
WHERE 'rank' = 1;
```

2.3.2 Algèbre relationnelle

Cette requête nécessite d'agréger les quêtes par niveau de difficulté et de sélectionner celles ayant la récompense maximale par groupe. Cela requiert des opérateurs comme GROUP BY qui n'existent pas en algèbre relationnelle.

2.3.3 Calcul tuple

Cette requête ne peut pas être exprimée en calcul relationnel tuple, car ce formalisme est basé sur la théorie des ensembles et n'inclut ni ordre, ni limite.

2.4 Le PNJ possédant l'inventaire contenant les objets dont la valeur en or cumulée est la plus importante

2.4.1 SQL

```
SELECT npc.Name, SUM(i.Price * npcII.Quantity) AS TotalValue
FROM NPCs npc
JOIN NPCItemInventories npcII ON npc.Name = npcII.NPCName
JOIN Items i ON npcII.ItemName = i.Name
GROUP BY npc.Name
ORDER BY TotalValue DESC LIMIT 1;
```

2.4.2 Algèbre relationnelle

Cela repose sur une somme cumulée suivie d'une sélection du maximum, ce qui nécessite des agrégats et des comparaisons globales, qui sont absents de l'algèbre relationnelle.

2.4.3 Calcul tuple

Cette requête ne peut pas être exprimée en calcul relationnel tuple, car ce formalisme est basé sur la théorie des ensembles et n'inclut ni ordre, ni limite.

2.5 Le type d'objet (arme, armure, potion ou relique) le plus souvent offert en récompense de quêtes de niveau 5

2.5.1 SQL

```
SELECT item.Type, COUNT(*) as apparitions
FROM Items item
WHERE item.Name in (
    SELECT reward.ItemName
    FROM Rewards reward
    WHERE reward.ItemName <> 'Or' AND
        reward.QuestName in (
            SELECT quest.Name
            FROM Quests quest
            WHERE quest.Difficulty = 5
        )
)
GROUP BY item.Type
ORDER BY apparitions desc LIMIT 1;
```

2.5.2 Algèbre relationnelle

Il s'agit d'un GROUP BY sur le type d'objet filtré par niveau, suivi d'un COUNT(*). Aucun de ces opérateurs n'est défini dans l'algèbre relationnelle, donc c'est impossible.

2.5.3 Calcul tuple

Cette requête ne peut pas être exprimée en calcul relationnel tuple, car ce formalisme est basé sur la théorie des ensembles et n'inclut ni ordre, ni limite.

2.6 Les monstres avec les meilleures récompenses en valeur en or cumulée en fonction de leurs points de vie

2.6.1 SQL

```
SELECT (MonsterCost/MonsterHealth) AS Ratio, Name
FROM (
    SELECT SUM(ItemPrice * lootQuantity) as MonsterCost, MonsterName
    FROM
    (
        SELECT
            items.Name as ItemName,
            items.Price as ItemPrice,
            loot.MonsterName as MonsterName,
            loot.Quantity as lootQuantity
        FROM Items items
        JOIN MonsterLoots loot ON items.Name = loot.LootName
    ) as subrequest
    GROUP BY MonsterName
) as monsterHealthAndPrice
```

```
JOIN Monsters monsters on monsters.Name = MonsterName  
ORDER BY Ratio;
```

2.6.2 Algèbre relationnelle

Cette requête implique un groupement par points de vie, puis une comparaison. C'est impossible dans le formalisme de l'algèbre linéaire.

2.6.3 Calcul tuple

Cette requête ne peut pas être exprimée en calcul relationnel tuple, car ce formalisme est basé sur la théorie des ensembles et n'inclut ni ordre, ni classement.