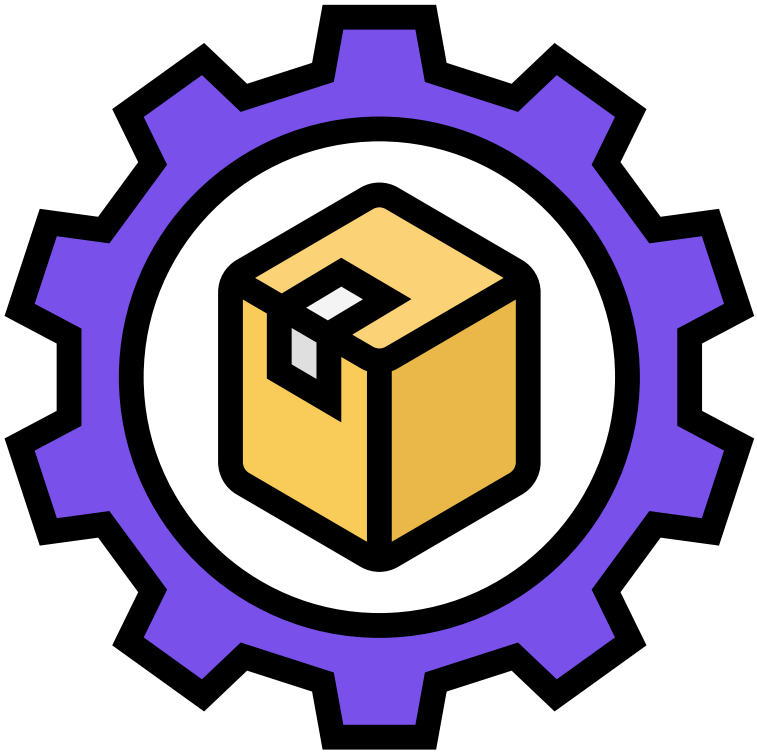
Modélisation et stockage de données GitHub dans une base de données relationnelle

PAR: YASSINE HARRATI

link: https://github.com/yaserrati/analyse-data-github/tree/main

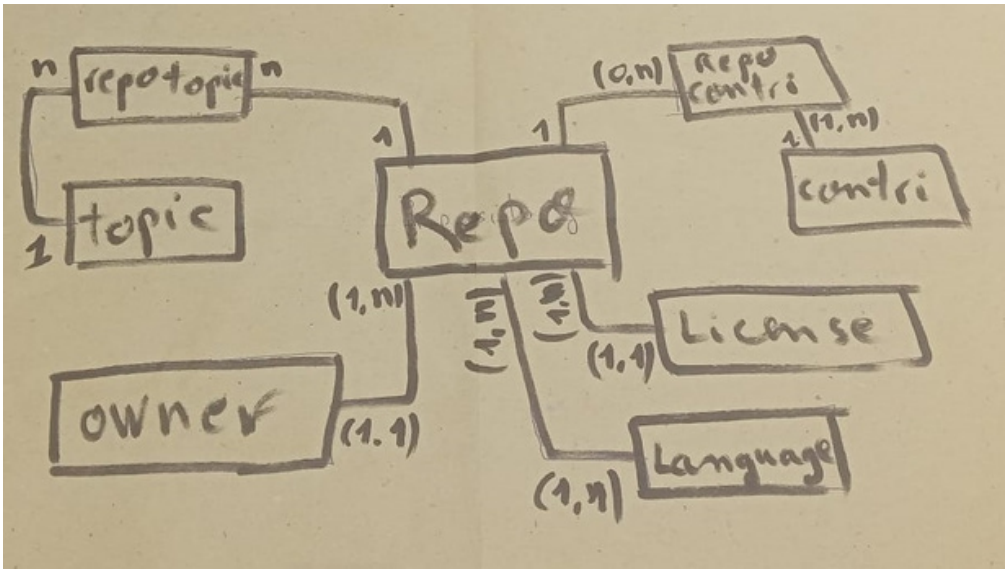
Introduction :

Le but de ce projet était de modéliser et de stocker des données de GitHub dans une base de données SQL pour permettre une analyse plus approfondie. La base de données servira de référentiel centralisé pour toutes les informations extraites, facilitant ainsi l'accès, la recherche et la manipulation des données.

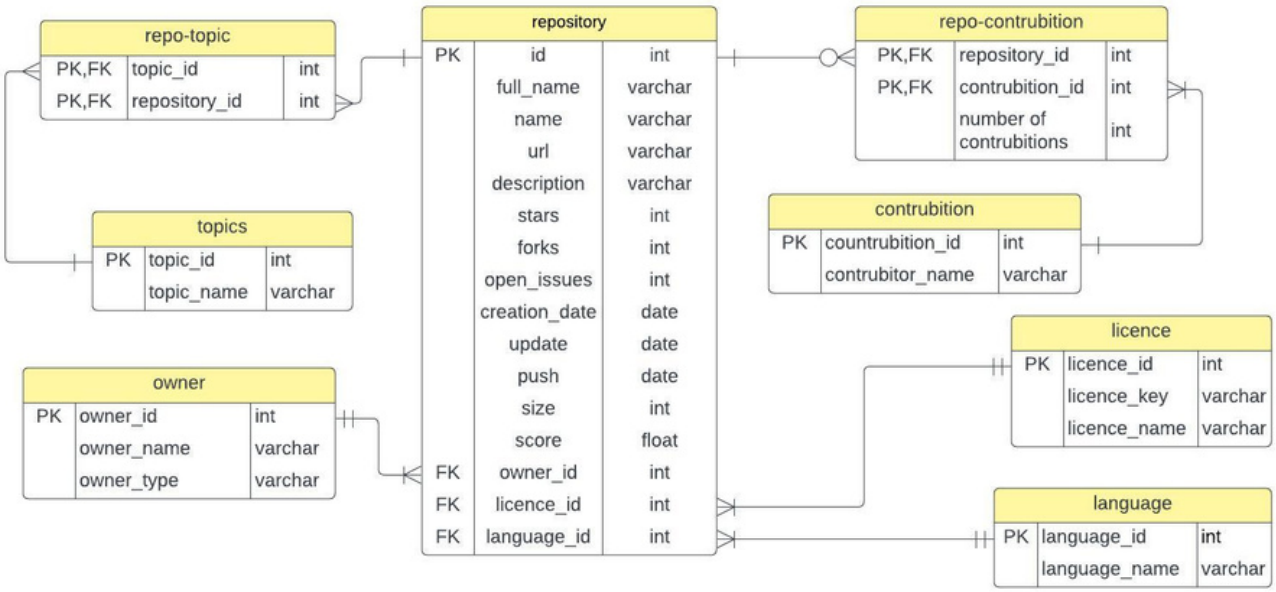


Modélisation :

Nous avons commencé par modéliser notre base de données à l'aide d'UML. Nous avons identifié 8 tables nécessaires pour stocker toutes les informations pertinentes extraites de GitHub : language, Contribution, Licence, Owner, Topics, Repository, Repo-Contribution et Repo-Topic..

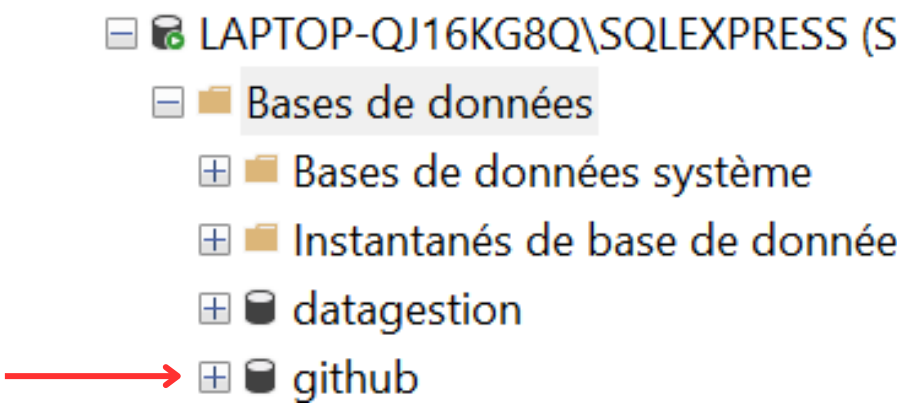


Ensuite, nous avons créé un diagramme ERD pour visualiser les relations entre ces tables et pour nous aider à concevoir notre base de données. Le diagramme ERD a montré les clés primaires et étrangères ainsi que les relations entre les tables.



Création de la base de données :

Nous avons créé la base de données sur SQL Server en utilisant le script fourni par notre équipe de développement. Le nom de la base de données est "github". Nous avons créé les 8 tables nécessaires en utilisant les spécifications de notre modèle UML et ERD.



Création des tables dans la base de données :

J'ai créé les tableaux qui se trouvent dans conception de modélisation

Insertion des données :

Après avoir créé les tables, nous avons inséré les données extraites de GitHub dans la base de données. Nous avons utilisé des scripts Python pour extraire les données de GitHub sous forme de fichiers CSV, puis nous avons utilisé SQL Server Management Studio pour importer les fichiers CSV dans nos tables.

Indexation :

Nous avons créé des index appropriés pour améliorer les performances de recherche et d'interrogation dans notre base de données. Nous avons créé des index pour les colonnes les plus souvent utilisées dans les requêtes, et nous avons également créé des index pour les colonnes qui ont des données uniques.

```
CREATE INDEX ownerIndex on "owner" ("owner_name")

CREATE INDEX topicsIndex on "topics" ("topic_name")

CREATE INDEX repoIndex on repository ("id")

CREATE INDEX licenseIndex on licence ("licence_name")

CREATE INDEX contributorIndex on contribution ("contrubitor_name")
```

Vérification de la qualité et de la rapidité des index :

Nous avons créé des requêtes SQL pour vérifier la qualité et la rapidité des index que nous avons créés. Nous avons exécuté des requêtes de recherche pour voir si les index ont amélioré les performances de la recherche et nous avons également exécuté des requêtes de jointure pour voir si les index ont amélioré les performances de l'interrogation
Les résultats ont montré une nette amélioration des performances de recherche dans la base de données.

Conclusion :

Ce projet nous a permis de modéliser et de stocker efficacement les données GitHub dans une base de données relationnelle. Les index créés ont grandement contribué à améliorer les performances de recherche. Ce référentiel centralisé facilitera l'accès, la recherche et la manipulation des données pour les analyses futures.