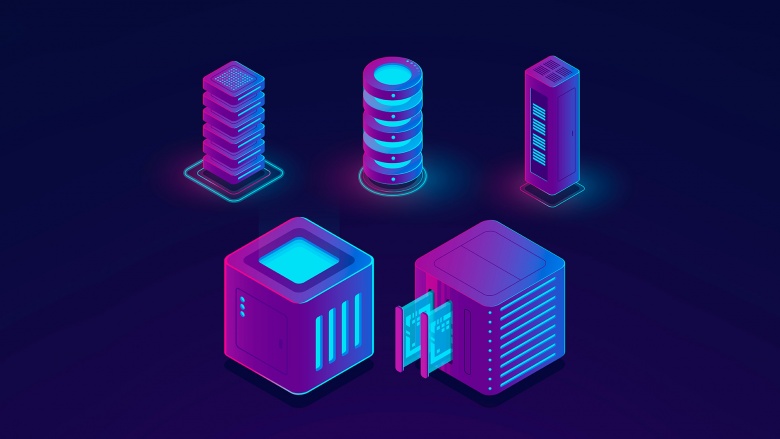
**Rapport de Projet :**

**Modélisation et Stockage des Données GitHub dans MySQL**



**Réalise par : TAHA SRHAYAR**

**Encadré par : Moad Tajmouati**

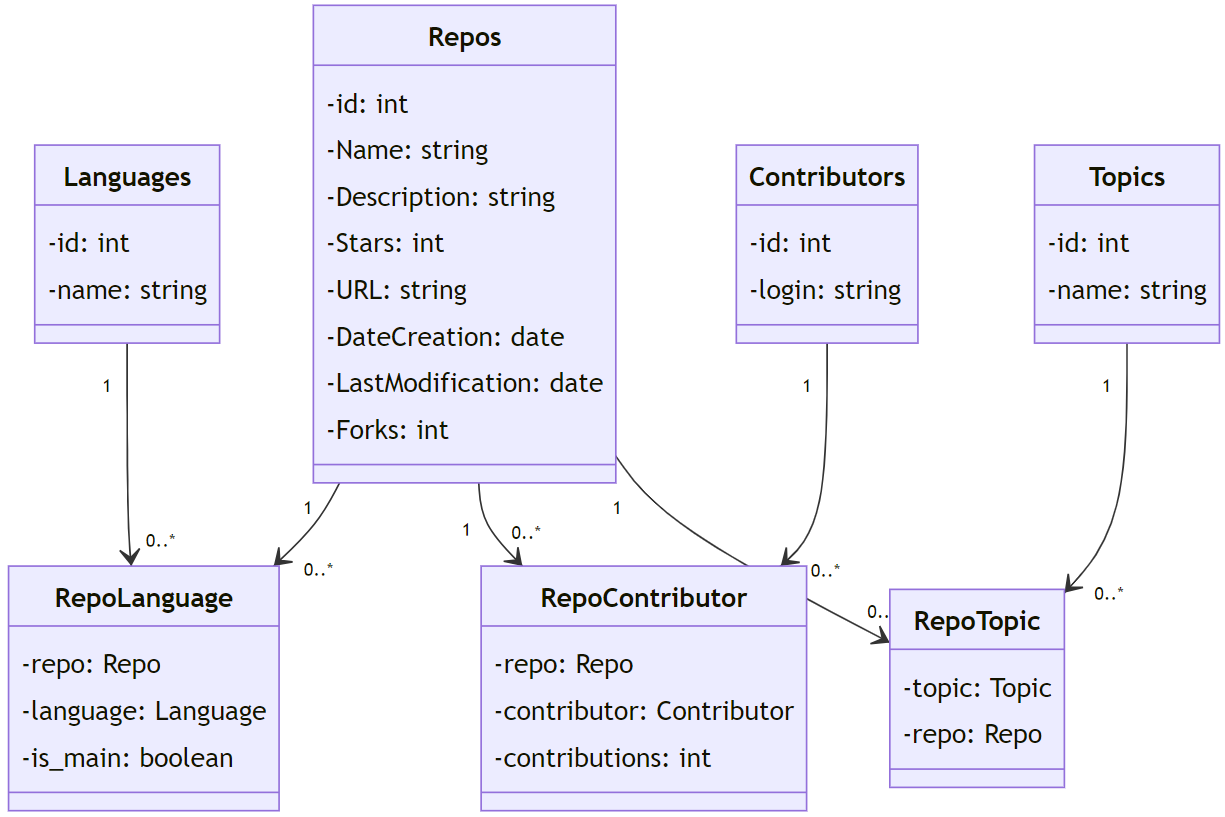
**Introduction :**

En tant que développeur Data, l'objectif principal de ce projet était de collecter, modéliser et stocker les données extraites de GitHub dans une base de données MySQL. Cette base de données servira de référentiel centralisé pour toutes les informations collectées, facilitant ainsi l'accès, la recherche et la manipulation des données. L'analyse de l'écosystème GitHub sera grandement améliorée grâce à cette base de données bien structurée.

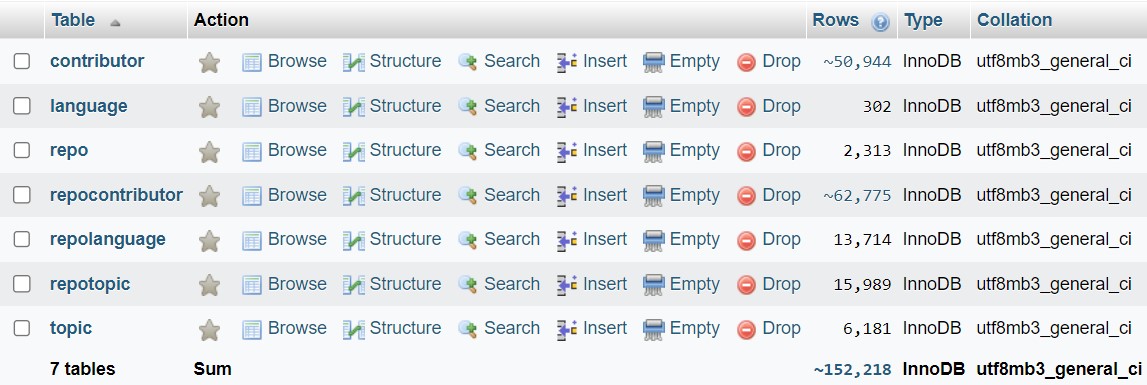
**Modélisation de la Base de Données :**

La base de données comprend sept tables clés qui jouent chacune un rôle essentiel dans la gestion et l'organisation des données :

*Digramme de classe*



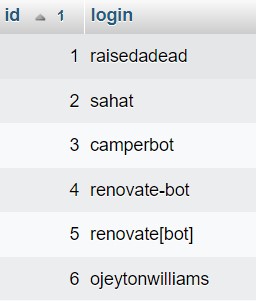
*Base de données (github)*



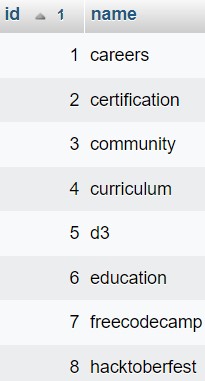
1. **Table "Repos"** : Cette table stocke les détails des référentiels GitHub, tels que leur nom, leur description, le nombre d'étoiles, l'URL, les dates de création et de dernière modification, ainsi que le nombre de forks. La colonne `id` sert de clé primaire pour identifier chaque référentiel de manière unique.



1. **Table "Contributors"** : Dans cette table, nous stockons les informations sur les contributeurs, en enregistrant leur identifiant et leur nom d'utilisateur (login). La colonne `id` est la clé primaire pour identifier chaque contributeur.



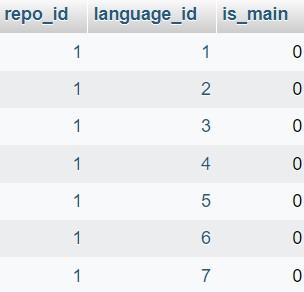
1. **Table "Topics"** : Cette table garde une trace des différents sujets (topics) associés aux référentiels, en stockant leur identifiant et leur nom. La colonne `id` est la clé primaire.



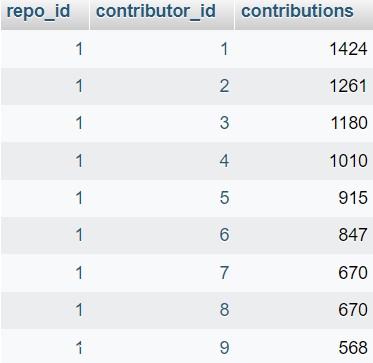
1. **Table "Languages"** : Ici, nous enregistrons les langages de programmation utilisés dans les référentiels, en stockant leur identifiant et leur nom. La colonne `id` est la clé primaire.



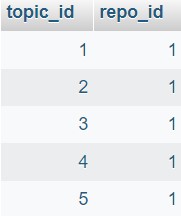
1. **Table "RepoLanguage"** : Cette table établit une relation de nombreux-à-plusieurs entre les référentiels et les langages. Elle stocke les clés étrangères des référentiels et des langages liés, ainsi qu'une indication sur le langage principal du référentiel avec la colonne `is\_main`.



1. **Table "RepoContributor"** : De manière similaire, cette table établit une relation de nombreux-à-plusieurs entre les référentiels et les contributeurs. Elle stocke les clés étrangères des référentiels et des contributeurs associés, ainsi que le nombre de contributions de chaque contributeur au référentiel.



1. **Table "RepoTopic"** : Enfin, cette table établit une relation de nombreux-à-plusieurs entre les référentiels et les sujets (topics). Elle stocke les clés étrangères des référentiels et des sujets liés.



**Processus d'Extraction et de Stockage des Données**

Le processus d'extraction et de stockage des données est divisé en plusieurs étapes :

1. **Extraction des Données** : Les données sont extraites de GitHub à l'aide d'un processus de collecte bien défini. Les informations clés telles que les détails des référentiels, les contributeurs, les sujets et les langages sont collectées de manière exhaustive.
2. **Transformation des Données** : Les données extraites sont ensuite nettoyées et transformées pour correspondre au schéma des tables dans la base de données. Les valeurs manquantes sont gérées de manière appropriée, et les formats de date sont convertis au format requis pour la base de données.
3. **Chargement des Données** : Les données nettoyées sont ensuite chargées dans la base de données MySQL à l'aide du langage SQL. Les requêtes d'insertion sont optimisées pour un chargement efficace et rapide des données.
4. **Vérification et Validation** : Une fois les données chargées, des requêtes SQL sont utilisées pour vérifier l'intégrité et la validité des données stockées dans la base de données. Les contraintes de clé étrangère sont vérifiées pour assurer la cohérence des relations entre les tables.

**Véracité des Données et Incertitude des Mesures :**

La véracité des données et l'incertitude des mesures sont deux aspects essentiels pris en compte lors de ce projet. Pour garantir la véracité des données, nous avons veillé à collecter des informations à partir de sources fiables et crédibles. Les données ont été soigneusement nettoyées pour éliminer les entrées incorrectes ou incohérentes.

Cependant, il est important de noter que les données de GitHub sont continuellement mises à jour par la communauté des développeurs. Par conséquent, certaines mesures telles que le nombre de stars, le nombre de forks ou le nombre de contributions peuvent changer au fil du temps. L'incertitude de ces mesures doit être prise en compte lors de l'analyse des données et de l'interprétation des résultats.

**Conclusion :**

Ce projet de modélisation et de stockage des données de GitHub dans une base de données MySQL a été un succès. La base de données bien structurée fournit un référentiel centralisé pour toutes les informations collectées, ce qui facilite l'analyse approfondie de l'écosystème GitHub.

En prenant en compte la véracité des données et l'incertitude des mesures, les résultats obtenus lors des futures analyses à partir de cette base de données seront fiables et informatifs. Ce projet a ouvert la voie à de futures recherches et analyses visant à mieux comprendre la communauté GitHub et ses référentiels.