

Big Data : ingénierie et traitement

Projet de groupe

1. Consignes

Le projet de groupe sera divisé en trois parties :

1. La rétro-ingénierie de différentes bases de données (SQL et NoSQL) déployées physiquement sur différentes machines, et liées entre elles.
2. La création et l'exécution de requêtes SQL et NoSQL sur ces bases de données afin de récupérer des informations présentes sous différentes formes et dans différentes bases de données physiques.
3. L'élaboration d'un schéma physique représentant le même domaine d'application, amélioré et évolué suivant certaines contraintes.

1.1. Rétro-ingénierie des bases de données

Sur base d'un ensemble de bases de données déployées pour vous, et sur base d'un schéma conceptuel, il vous est demandé d'élaborer un schéma physique représentant la répartition physique des données dans les différentes tables, collections, paires de clé-valeur, ...

Pour ce faire, vous aurez à disposition :

- Le schéma conceptuel représentant les types d'entités et les types d'associations du domaine d'application.
- L'accès aux bases de données physiques dans lesquelles les données sont stockées, et structurées différemment pour chaque groupe.

Sur base des données stockées et du schéma conceptuel, il vous est demandé de réaliser cette tâche de rétro-ingénierie afin de construire un schéma physique du domaine d'application, montrant comment les types d'entités et types d'associations du schéma conceptuel sont représentés et répartis physiquement dans les différentes bases de données.

1.2. Création et exécution de requêtes

Il vous est demandé de générer des requêtes sur les différentes bases de données afin de récupérer des informations stockées parfois dans une seule et même structure physique, parfois dans différentes structures physiques voir même différentes bases de données.

En particulier, vous devez :

- Créer des requêtes dans les différents langages propres aux bases de données (SQL, MongoDB, Redis, ...),
- Récupérer les résultats des requêtes afin de les afficher, ou de les utiliser comme input pour une autre requête (dans le cas où vous devez récupérer des données dans différentes bases de données). Pour cela, il vous est conseillé de réaliser un mini-programme dans le langage

de programmation de votre choix (Java ou Python si possible) qui se connectera à vos différentes bases de données, et qui exécutera des requêtes sur celles-ci.

- Afficher les requêtes et les résultats finaux, ainsi que le temps d'exécution des différentes requêtes (un simple message dans la console suffit).

Cette étape vous permettra de vous rendre compte de la complexité de créer des requêtes pour interroger vos données stockées dans des structures physiques et bases de données différentes. Parfois des résultats de requêtes devront être utilisés comme input pour d'autres requêtes dans un autre langage. La manipulation et la transformation de données intermédiaires peut être complexe et surtout coûteuse dans certains cas, d'où l'importance de bien structurer vos données en fonction des besoins de votre application : performance (en lecture et/ou en écriture), disponibilité, cohérence des données, tolérance aux erreurs, facilité d'écriture de requêtes, diversité des langages de requête, ...

1.3. Elaboration d'un schéma physique amélioré

Sur base des résultats de vos requêtes, il vous est demandé de réfléchir à une restructuration physique de vos bases de données afin d'améliorer certains critères propres à chacun des groupes. Il vous sera par exemple demandé d'améliorer les temps d'exécution des requêtes en lecture seulement en prévoyant une importante augmentation des données stockées dans vos bases de données. Ou encore minimiser le nombre de requêtes pour accéder aux mêmes données, sans supprimer de bases de données tout en prévoyant que ces requêtes seront de plus en plus souvent exécutées à la suite d'une augmentation de l'utilisation de l'application.

Vous pouvez restructurer vos structures physiques de vos bases de données sur différents points :

- Ajout/migration de bases de données
- Fusion/découplage/imbrication
- Normalisation/dénormalisation
- Augmentation du nombre de nœuds pour répartir les données
- Réplication de données
- ...

En plus de ces recommandations d'évolution du schéma physique, il vous sera demandé de réfléchir à quelques mises en situation proposées à chaque groupe, afin de répondre à différents problèmes liés aux bases de données et à leur utilisation, dans un contexte Big Data afin de tester vos connaissances sur la matière.

2. Remise du travail

Votre projet est à remettre au plus tard le **lundi 22 mai 2023 à midi** par mail à l'adresse suivante : a_ouared@esi.dz.

L'évaluation du travail portera sur le rapport répondant aux trois parties de l'énoncé, ainsi que sur celle de la démonstration finale. Cette dernière aura lieu pour chaque groupe le jour de l'examen. Chaque groupe se verra attribué un évaluateur (assistant ou professeur) et sera évalué par ce dernier le jour de la démonstration. Cette évaluation représentera votre note finale pour ce cours pour la session de juin.

La démonstration consistera en une présentation en présentiel des résultats obtenus de chacune des trois parties (présentation du schéma physique à la suite de la phase de rétro-ingénierie, exécution des requêtes pour la seconde partie, et argumentation sur le schéma physique amélioré ainsi que sur les mises en situation pour la dernière partie).

Il vous est demandé de former des groupes de 5-6 personnes le plus rapidement possible pour vous lancer dès que possible dans le projet.

N'hésitez surtout pas à me contacter régulièrement par mail ou via Teams pour toute question ou prise de rendez-vous.

3. Liens utiles

- Lien pour télécharger gratuitement RedisInsight :
<https://redis.com/fr/redis-enterprise/redisinsight/#insight-form>
- Lien pour télécharger gratuitement DataGrip :
<https://www.jetbrains.com/datagrip/download>
- Documentation sur l'API Python permettant de se connecter et communiquer avec une base de données MySQL :
<https://pypi.org/project/mysql-connector-python/>
<https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html>
- Documentation sur l'API Java permettant de se connecter et communiquer avec une base de données MySQL :
<https://waytolearnx.com/2020/05/connexion-a-une-base-de-donnees-mysql-avec-jdbc-java.html>
- Documentation sur l'API Python permettant de se connecter et communiquer avec une base de données MongoDB :
https://www.w3schools.com/python/python_mongodb_getstarted.asp