Projet

1. Présentation

Le projet de l'UE Bases de Données vous permettra de reprendre toutes les connaissances et compétences acquises ou renforcées dans les différentes EC traitant des bases de données sur des données réelles. En effet, ces données qui vont être utilisées sont celles du Metropolitan Museum of Art et concernent plusieurs centaines de milliers d'objets d'art.
Le projet consiste, par équipe de 4 personnes maximum, à créer une base de données, un ensemble de requêtes, de vues, de transactions et de triggers afin de fiabiliser les données et de faciliter le travail des chercheurs travaillent sur ces données. Pour cela, il vous faudra observer les données, les comprendre, les corriger, les organiser dans une base de données normalisée et optimisée et proposer un ensemble de requêtes pour gérer ces données le plus efficacement possible.

Les données sont fournies en format CSV avec de nombreuses colonnes (plus de 50), le fichier est assez propre mais il subsiste quelques problèmes qu'il faudra régler. Attention le fichier est volumineux, il n'est pas question de tout corriger pour obtenir une base parfaite mais de montrer, à travers ce que vous allez faire que vous maitrisez les différentes étapes.
https://github.com/metmuseum/openaccess

La question centrale à laquelle les chercheurs veulent répondre est la suivante :
« **comment évolue l'utilisation de matériaux pour la conception d'œuvres au cours du temps et selon les lieux** ».

Votre travail sera suivi au second semestre d'un second projet sur le même thème mais sur d'autres aspects de gestion de bases de données, en particulier sur l'indexation et l'optimisation des requêtes et sur la conception d'une interface de type « tableau de bord » pour accéder aux données.

2. Consignes

Le rendu attendu pour le projet consiste en :

- Un unique fichier sql contenant toutes les instructions, clairement constitué, sans erreur (commentez les zones éventuelles qui bloquent).
- Un unique fichier pdf contenant toutes les explications supplémentaires (types de données, modèle entités-associations, etc.)

Les étapes à suivre sont les suivantes, tout doit être fait en SQL avec Postgresql :

1. **Import des données.**
   Remarques : vous ne devez pas modifier les données avant import, sauf pour régler des problèmes de format le cas échéant, et tout importer en bloc dans une seule grosse table sql.
   Rendu : la requête d'import et la structure de la table dans laquelle les données sont importées avec une justification des types de données quand ils sont non triviaux. S'il y a des modifications avant import elles doivent être limitées au maximum et justifiées.
2. **Nettoyage des données.**
   Remarques : le nettoyage des données ne doit pas être exhaustif mais doit permettre de régler plusieurs problèmes typiques dans les données, par exemple les problèmes de dates (voir colonne AC par exemple), la suppression des attributs inutiles pour le problème posé, etc.

Rendu : un ensemble de requêtes SQL (requêtes simples ou fonctions selon le cas) pour le nettoyage, la liste des attributs conservés en justifiant les choix, la structure de la table unique contenant les données nettoyées.

3. **Modélisation, normalisation et représentation avec un modèle entités-associations.**
   Uniquement pour la sortie de la partie 2 vous devez effectuer tout le processus de normalisation vu en cours et transférer les données depuis votre table unique nettoyée vers les différentes tables obtenues à l'issue du processus de normalisation. Le transfert risque de nécessiter quelques requêtes et fonctions supplémentaires.
   Rendu : un modèle entités-associactions, la structure des tables une fois la normalisation effectuée, les requêtes/fonctions de transfert.
4. **Exécution de requêtes.**
   Vous devez proposer un ensemble de requêtes, vues, tables… pertinentes pour répondre à la question en gras de la partie 1 mais vous pouvez également répondre à des questions proches. Il peut, par exemple, être intéressant de stocker des statistiques dans des tables supplémentaires pour éviter de refaire les calculs à chaque exécution…
   Rendu : un ensemble de requêtes SQL, vues... avec des explications sur leur pertinence.
5. **Conception de triggers.**
   Vous devez écrire quelques triggers pour assurer des vérifications en lien avec la base (par exemple mise à jour des tables de statistiques en cas d'insertion ou de modifications, vérification de l'intégrité de la base…). Le choix est libre mais doit être justiié.
   Rendu : un ensemble de triggers.

3. Dépôts

Vous pouvez et c'est même conseillé déposer avant la date limite) :

1. **date limite le 10 décembre à 23h55 – un unique fichier compressé contenant :**
   - Rapport-nom1-nom2-nom3-nom4.pdf : justifiant tous les prétraitements effectués sur les données des parties 1 et 2.
   - Rapport-nom1-nom2-nom3-nom4.sql : contenant toutes les instructions sql associées.
2. **date limite le 19 décembre à 23h55 – un unique fichier compressé contenant :**
   - Rapport-nom1-nom2-nom3-nom4.pdf : le document précédent en ajoutant tous les éléments liés à la normalisation, partie 3.
   - Rapport-nom1-nom2-nom3-nom4.sql : le document précédent en ajoutant toutes les instructions sql associées à la normalisation.
3. **date limite le 07 janvier à 23h55 – un unique fichier compressé contenant :**
   - Rapport-nom1-nom2-nom3-nom4.pdf : le document précédent en ajoutant toutes les explications sur les requêtes/triggers/tables, parties 4 et 5.
   - Rapport-nom1-nom2-nom3-nom4.sql : le document précédent en ajoutant toutes les requêtes/triggers/tables.

4. FAQ

- **Est-il possible de faire des groupes de plus de 4 personnes ?** Non !
- **Est-il possible de faire des groupes de moins de 4 personnes ?** C'est déconseillé.
- **Doit-on déposer les fichiers pour chaque personne ?** Non, un par groupe suffit.
- **Combien de requêtes faut-il faire ?** Assez pour montrer que vous avez compris comment faire des requêtes on triviales, des triggers pertinents et le tout en lien avec le sujet. Si en plus elles sont différentes de celles des autres groupes… Si vous en avez moins de 5 c'est probablement insuffisant, plus de 20 c'est probablement trop.
- **Faut-il faire une interface ?** Non, ce sera au semestre 6.
- **Peut-on utiliser autre chose que postgresql ?** Non !