

Projet collections d'objets culturels

Il vous est demandé de réaliser la base de données pour un site ou une application de gestion de collections d'objets culturels. Les utilisateurs pourront répertorier les différents objets culturels qu'ils possèdent ou qu'ils ont possédés (livres, films, jeux vidéo, ...), de les noter/commenter et de les organiser par listes thématiques.

Les informations suivantes sont conservées pour chaque utilisateur : son nom, son prénom, son adresse et sa date de naissance. La date d'inscription est enregistrée. Un login et un mot de passe sont demandés à l'inscription. Le login est composé de la première lettre du prénom et des 7 premières lettres du nom (en minuscules) suivies de deux chiffres. Les logins de plusieurs adhérents différents ne peuvent être identiques. Le mot de passe ne peut contenir que des lettres (minuscules/majuscules), des chiffres et le caractère '_'.

Un utilisateur peut aussi créer une liste d'objets (les objets culturels d'une liste doivent être de même type, c'est à dire soit tous des livres, soit tous des films, etc.). Un nom est associée à chaque liste. Il est possible d'ajouter un descriptif à la liste et à chaque objet de la liste, afin de justifier leur présence dans la liste). Chaque objet est normalement présent une seule fois dans la base, mais peut appartenir aux listes (aussi appelées collections) de plusieurs utilisateurs. Un utilisateur peut aussi prévoir d'acheter des objets culturels.

Un utilisateur peut noter (sur une note de 1 à 20) chacun des objets de ses collections.

Partie 1, à faire individuellement (date limite : 4 novembre)

Modélisation

Dessiner le modèle entité-association de la base de données pour les collections d'objets en respectant au maximum les besoins fonctionnels indiqués dans l'énoncé.

Préciser les contraintes d'intégrité sur le modèle (contraintes que les données devront respecter à tout instant).

Les contraintes sont indiquées en version texte (paragraphe contenant la liste des contraintes qui seront à implémenter, c'est à dire à exprimer en SQL)

Établir le modèle logique relationnel de la base (l'ensemble des relations déduites du modèle entité-association).

Une relation sera de la forme $R(X_1 : T_1, X_2 : T_2, \dots, X_N : T_N)$ avec R nom de la relation, X_i les noms des attributs et T_i les types abstraits (par exemple : `String` plutôt que `Varchar`). Ne pas oublier de préciser les clés primaires/clés étrangères.

Rendu du projet : un rapport devra être remis contenant le schéma E/A, la liste des contraintes d'intégrité ainsi que le modèle relationnel.

Implémentation de la base de données

Rendu du projet :

Les scripts SQL de création des tables, de suppression des tables et d'insertion de données de tests devront être remis.

Les contraintes d'intégrités statiques (pouvant être implémentées sans utiliser de PL/SQL) seront aussi exprimées en SQL dans les scripts de création de tables.

Un script permettra d'exécuter tous les autres scripts afin de, si nécessaire, recréer les tables et de remplir les tables.

Partie 2 (à faire en binôme ou individuellement, date limite : 16 décembre)

Requêtes SQL

Ecrire les requêtes permettant d'obtenir les résultats suivants :

1. Les utilisateurs ayant créé une liste pour chacun des types d'objets culturels existants.
2. Les objets culturels appartenant à plus de 20 collections et dont la moyenne des évaluations est supérieure à 14.
3. Les utilisateurs n'ayant jamais mis une note inférieure à 8 à un objet culturel.
4. L'objet culturel le plus commenté au cours de la dernière semaine.
5. Pour chaque utilisateur ayant au moins noté des objets sur 3 mois consécutifs au cours de l'année précédente (par exemple, une note a été ajoutée en juin, en juillet et en août), indiquer son nombre d'objets possédés, son nombre d'objets à acheter, la longueur minimale, maximale et moyenne de ses collections.

Définir les index pour optimiser vos requêtes.

Procédures et fonctions PL/SQL

1. Définir une fonction qui retourne pour une œuvre dont l'identifiant est passé en paramètre et si elle a plus de 20 évaluations, le score moyen des notes pour cette œuvre. Si l'œuvre a moins de 20 appréciations ou si elle n'existe pas, la fonction renverra un score nul.
2. Ecrire une procédure qui génère pour un utilisateur la liste de ses 10 films préférés, la liste de ses 10 livres préférés et la liste de ces 10 jeux vidéos préférés. Pour chacun des objets, en cas d'égalité, la priorité sera donnée aux objets dont l'ajout est le plus récent. Si une liste n'a pas au moins 10 éléments elle ne sera pas générée.
3. Ecrire le code PL/SQL permettant de générer des suggestions pour un utilisateur, c'est dire les X objets dont le score moyen est le plus élevé pour les objets communs à au moins Y utilisateurs, ces utilisateurs ayant mis la même note que l'utilisateur courant pour au moins Z objets (ces objets culturels ne sont pas déjà dans une liste de l'utilisateur).
Il s'agit d'identifier les objets de personnes ayant des goûts similaires que l'utilisateur courant, et de prendre les objets les mieux évalués de cet ensemble.

Déclencheurs

Définir les déclencheurs associés à la base.

En particulier, définir les déclencheurs suivants :

1. Définir les listes des ajouts du mois 'xxx' pour l'année 'yyy' pour un utilisateur « nouveauté ».
Chaque nouvel objet culturel ajouté dans la base de données au courant du mois 'xxx' de l'année 'yyy' est ajouté à la liste correspondante.
2. Archiver les listes supprimées dans une table d'archivage.

Rendu du projet

Vous devez déposer sur Moodle (au plus tard le 16 décembre) :

- les requêtes SQL,
- les définitions de contraintes d'intégrité,
- les scripts (fonctions, procédures, déclencheurs) PL/SQL,
- les index,
- si nécessaire définir les débuts/fins de transaction pour vos opérations,
- un script permettant de tester les procédures/fonction/déclencheurs.