

Rapport BDD2

Introduction

Dans le cadre de l'UE «Base de données 2» il m'a était demandé de réaliser un projet sur un site de diffusion de vidéos en replay. C'est un outil que j'utilise assez souvent chez moi donc c'était assez simple de visualiser ce qu'il fallait faire dans les grandes lignes. Mais c'est toujours intéressant de pouvoir «coder» un outil que l'on utilise souvent dans sa vie quotidienne. Généralement à l'Université on fait des projets sur des situations simulés mais là ça touche à quelque chose d'assez connu et de courant pour un peu tout le monde.

J'ai donc dû implémenter les tables de données de ce site de replay, mais aussi gérer les différentes contraintes et faire quelques requêtes et bloc PL/SQL.

Planification

Le sujet nous imposés de faire la première partie seul : c'est à dire le schéma relationnel, le modèle entité-association, les tables et les requêtes.

Pour la seconde partie qui comportée les contraintes et les blocs PL/SQL on pouvait la faire en binôme mais j'ai décidé de la faire seul.

Dans la suite vous trouverez les explications sur les différents fichiers sql ainsi que les explications sur le modèle entité-association et le schéma relationnel.

Le modèle entité-association

Afin d'illustrer mes tables d'une façon très claire, j'ai décidé de mettre beaucoup d'id en clés étrangères et donc en clés primaires de mes tables principales. Les tables qui sont donc centrales pour la réalisation de ce projet sont : Utilisateur, Vidéo, Archives.

Les tables dites secondaire sont : Historique, Favoris, Abonnement, Suggestions, emission, categorie_emission, categorie.

Et enfin troisièmement Newsletter qui est pour moi pas très importante sur un site de replay.

J'ai donc construit ce schéma autour de vidéo essentiellement.

Les tables favoris,suggestions,abonnement et utilisateur_categorie n'ont rien de propre à leur table c'est pour cela que l'on peut les représentés comme des associations mais j'ai préféré les représenté comme des tables comme les autres.

Tout est fait dans mon schéma pour qu'avec un maximum de 4 jointures on puissent accéder à tout.

Pour expliquer les relations je peut dire que un utilisateur à plusieurs vidéos favorites, ainsi que plusieurs suggestions qui lui sont faites , et qu'il a peut-être plusieurs catégories d'émissions qui lui plaise. Enfin un utilisateur peut regarder plusieurs vidéos.

Pour expliquer les relations de vidéos, je peut dire que un id_de vidéo peut être archiver plusieurs fois car si on supprime une vidéo son id est supprimer et donc archiver et qu'il y aura le même id associé a un autre nom et une autre emission dans la table vidéo peut être plus tard, donc id_archive est bien la clé primaire de archives , et il n'y a donc pas de clés étrangères dans archives comme j'ai pu le mentionner dans le premier rendu.

Une vidéo peut donc être associé à une seule émission et une émission peut comporter plusieurs vidéos qui seront donc des épisodes. Chaque émission possède une et une seule catégorie.

Pour ce qui est du 3ème échelon , la Newsletter , je ne savais pas trop quelle relation mettre, car on dit qu'un utilisateur s'abonne à une newsletter. Je pense que ma première idée qui était de mettre les id d'utilisateur dans la table newsletter était une bonne idée mais j'ai préféré simplifier la chose en mettant un id_news, comme ça chaque semaine on a une newsletter précise et donc on pourrait tenir une autre table qui signifie si un utilisateur est abonné à la newsletter de la semaine.

Pour finir je dirais qu'il me manque la donnée sur la date de disponibilité , je ne savais pas trop où la placé , beaucoup d'hésitation entre vidéo et émission. C'est pour cela que la contrainte dynamique numéro 2 n'a pas était implémentée.

Implémentation

1) Les tables

Utilisateur et Vidéo sont très détaillés dans le sujet et j'ai donc décidé de suivre les règles du sujet de mettre le nom, la date de diffusion, le pays , la durée, le multilingue ainsi que les id pour les associations.

Idem pour utilisateur , j'ai décidé de conserver le login et le password qui était obligatoire même si je n'utilise pas ses informations pour naviguer entre les tables. Enfin j'ai conservé l'email, le prénom , le nom , la date , et la langue car c'est ce qui fait que l'on va pouvoir reconnaître précisément un utilisateur et d'éviter les homonymes.

Les tables de second rang, favoris,abonnement,historique et suggestions sont des fonctionnalités du site de replay que l'utilisateur appréciera car cela lui permet un accès facile à ses vidéos et à son activité sur le site. Elles sont très petites en informations car on peut retrouver les informations qui leur sont associés en faisant jointures qu'il faut pour avoir le nom de la vidéo que l'utilisateur 'Jacques' à regarder 12 fois.

2) PL/SQL

a) Les contraintes

J'ai implémenté toutes mes contraintes avec des contraintes dynamiques.

Le premier trigger est assez simple , il compte juste le nombre de fois qu'apparaît l'id user dans la table favoris et si il apparaît plus de 300 fois alors on lève une erreur.

Le second trigger n'a pas était implémenté.

Le 3ème on créer juste un id supplémentaire dans notre table archive et on l'insère

avec les données de la vidéo que l'on viens de supprimée.

Enfin le 4ème check toutes les parties de la date de du temps, année, heure, min, jour, mois ,année et regarde précisément en partant de la dernière date insérer d'un utilisateur si les précédentes sont dans la même minute ou non si plus de 2 plus celle qui à la date la plus récente alors il y'a erreur. On suppose donc que l'on insère dans l'historique des dates dans l'ordre de la plus ancienne à la plus récente ce qui est logique.

b) Les procédures et fonctions

Je n'est pas implémentée la première instruction.

Pour la seconde et la dernière j'ai utiliser un curseur qui parcours le résultat de notre requête.

3) Les requêtes

Par rapport au premier rendu je me suis permis de modifier ma première requête qui n'était donc pas correcte , elle est maintenant corrigée.

Conclusion

En conclusion je peut dire que j'ai mis pas mal de temps pour pas grand-chose , car au final l'implémentation des triggers et des fonctions et procédure et assez rapide quand on a compris le système et aussi quand on apprend à utiliser les outils rapides comme les curseurs.

Même si je n'apprécie pas le système Oracle car je suis habitué à faire du sqllite et du MariaDb ou du mysql pour les sites web , j'avoue que apprendre les bases de données avec Oracle est quelque chose d'enrichissant même si je ne l'aurais pas fait de mon propre chef car je n'aime pas trop la façon de se dissocier du logiciel libre.

Pour finir je dirais que ce projet n'est pas une grande réussite mais que l'aurais au moins terminé en partit.