Bases de données relationnelles 2022-2023
Professeur : Nastaran Fatemi
Assistant : Christopher Meier

Labo 4 : Opérations avancées SQL

1 Objectif

Il s'agira de mettre en pratique vos connaissances concernant les opérations avancées en SQL.

1.1 Indications

Utiliser la base de données *Pagila* (du Labo 3) pour effectuer les différentes opérations.

1.2 A rendre

Vous devrez nous rendre une archive ZIP contenant le fichier SQL qui contient les triggers, vues et procédures stockées à réaliser

Le laboratoire sera comme d'habitude rendu sur Cyberlearn avant le: **Lundi 5 Décembre 2022 à 23h59**

2 Consignes générales

- Utiliser le language procédural psql (`plpsql`).
- Pour chaque exercice, montrer que l'opération s'est bien déroulée : Donner toutes les requêtes utilisées pour vérifier le bon fonctionnement.
- Répondez aux questions supplémentaires dans des commentaires.

3 Triggers (déclencheurs)

1. Utiliser un Trigger sur la table `payment` , pour qu'à chaque insertion, le paiement soit majoré de 8% et la date de paiement soit mise à jour à la date courante du serveur.
2. Créer une nouvelle table `staff_creation_log` avec les attributs `username` et `when_created` . Créer un Trigger pour insérer une nouvelle ligne dans `staff_creation_log` , après qu'un tuple est inséré dans la table `staff` .
3. Utiliser un Trigger qui permet de mettre à jour l'adresse email d'un membre du personnel de manière automatique à partir de son prénom et de son nom selon le format "*prénom.nom@sakilastaff.com*".

4 Vues

4. On aimerait envoyer des cartes du Nouvel An à l'adresse postale du personnel. On voudrait déléguer cette tâche à un employé, Franklin, mais pour des questions de protection des données personnelles, on voudrait donner accès à Franklin seulement aux numéros de téléphone des membres du personnel et de leurs adresses postales, mais pas leurs adresses e-mail (ou, d'ailleurs, les mots de passe). Créer une vue sur la table du personnel qui permettra à Franklin de réaliser la tâche demandée.
   Est-ce que Franklin pourra mettre à jour la base de donnée à travers cette vue?

5. On aimerait demander à Franklin d'envoyer un rappel par email à tous les clients qui ont du retard. Définir la vue qu'il a besoin pour effectuer cette tache. L'email envoyé à chaque client doit contenir le titre du film qu'ils n'ont pas rendu, ainsi que le nombre de jours de leur retard (il y a un email par film qui est en retard).
   Indications:
   - Quelques secondes de retard est considérée comme un jour de retard.
   - Un film n'est pas rendu si `return_date` est null.
   - Pour ajouter un nombre n de jour à une date: `date_col + n * INTERVAL '1 day'`.
   - Dans Postgres, soustraire deux date entre elles retourne un type interval duquel on peut extraire le nombre de jour.
6. Utiliser la vue créée au point 5. pour créer une nouvelle vue qui affiche les clients ayant plus de trois jours de retard.
7. On aimerait demander à Franklin de calculer le nombre de location par client. Définir la vue qu'il a besoin pour effectuer cette tache. Donner la requête pour afficher les 20 premiers clients avec le plus de locations (`customer_id`, `first_name`, `last_name`, `nb_locations`) en utilisant la vue définie.
8. Créer une vue pour calculer le nombre total de location par jour.
   Combien de locations sont effectués en 1er août 2005 ? Donner la requête SQL.
   Indication: La fonction `date_trunc` dans Postgres permet d'extraire la partie date d'un timestamp.

5 Procédures/Fonctions stockées

9. Créer une fonction qui calcule le nombre de films proposés par le magasin. La fonction prend en entrée l'ID de magasin et retourne en sortie le nombre de films proposés par le magasin. Utiliser la fonction pour afficher le nombre de films proposés par magasin 1 et 2. Confirmer la réponse avec une requête SQL.
10. Créer une procédure pour mettre à jour la date de la mise à jour de tous les films à la date d'exécution de la procédure. Quelle est la date de la mise à jour des films avant et après cette procédure ?