

## TP Postgres

On suppose qu'une bibliothèque gère une base de données dont le schéma est le suivant (les clés primaires des relations sont soulignées) :

- Client (NumClient, Nom, Prenom, Adresse, Tel, Compte) ;
- Films (NumFilm, Titre, Genre, Prix, NombreDVD) ;
- Personne (NumPers, Nom, Prenom).
- Locations (IdLocation, DateLoc, NbeJourLoc, Livraison, NumFilm, NumClient);
- Reservations (IdReservation, DateRes, NbJourRes, NumFilm, NumClient);
- Joue (NumPers, NumFilm);
- Realise (NumPers, NumFilm).

### Question :

1. Créer une base « tp\_postgres », créer un utilisateur « tp » et donner lui accès à la base « tp\_postgres » uniquement
2. Ecrire le script de créations de table sous Postgres (avec les clés étrangères)
3. Créer les tableaux de données, avec des exemples de données cohérentes (8 clients, 10 films, 5 personnes, 10 locations, 5 réservations, 5 joue, 5 réalise)
4. Ecrire les requêtes SQL, ainsi que les résultats selon vos propres données
  - a. Quelles sont les clients ayant emprunté le livre "Avengers" ?
  - b. Quelles sont les personnes qui ont joué et réalisé le même film?
  - c. Quelles sont les films qui n'ont jamais été loués ni réservés ?
  - d. Afficher le top 3 des films les plus populaires (location + reservation)
  - e. Ecrire la requête qui permet d'afficher les infos clients, le nombre de location effectué, le nombre de réservation effectué. ( Ecrire d'abord une fonction qui compte le nombre de location et une autre fonction qui compte le nombre de réservation effectué par chaque client)
5. On veut avoir 3 tables contenant 3 types de clients. (clients\_mini, clients\_medium, clients\_premium). On veut créer une fonction qui va répartir les clients dans cette table selon les règles suivants:  
clients\_mini : ayant loué+reservé moins de 2 films  
clients\_medium : ayant loué+reservé moins de 5 films  
clients\_premium : ayant loué+reservé plus de 5 films

Note : vous pouvez créer des vues intermédiaires (donner les scripts)