

Donnez les requêtes SQL qui répondent aux questions suivantes. Le schéma du résultat doit être *exactement* celui donné entre parenthèses.

1. Quels sont, dans l'ordre alphabétique de leur nom, les usagers qui n'ont aucun emprunt dans la base ? (numéro, nom, prénom)

numéro	nom	prénom
--------	-----	--------

30	Leroy	Lucas
----	-------	-------

28	Petit	Alexandre
----	-------	-----------

2. Quelles sont les oeuvres écrites par plusieurs auteurs ? (titre, annee, auteurs) utilisez la fonction "array_agg" pour le champ "auteurs" du résultat.

titre	annee	auteurs
-------	-------	---------

Germinie Lacerteux	1865	['Edmont', 'Jules']
--------------------	------	---------------------

3. Pour chaque auteur, donnez le nombre de fois qu'il ou elle "est emprunté" dans la base. Le résultat est trié selon ce nombre d'emprunts, de plus élevé au plus faible. (prenom, nom, nombre d'emprunts). On ne tiendra pas compte des auteurs jamais empruntés.

prenom	nom	nombre d'emprunts
--------	-----	-------------------

Victor	Hugo	65
--------	------	----

de Goncourt	Edmont	18
Albert	Camus	16
Alexandre	Dumas	15

Exercice 2 (11 pts)

On nous demande, pour des raisons de droit des données, de s'assurer que tous les usagers stockés dans notre base sont bien des emprunteurs.

1. (2 pts) Faites une fonction booléenne qui prend en entrée un numéro d'utilisateur, et retourne Vrai si et seulement si cet usager a un emprunt.
2. (2 pts) A l'aide de cette fonction faite une requête qui affiche, pour chaque usager, s'il a un emprunt ou non. (numéro, a un emprunt)
3. (2 pts) Supprimez les usagers qui ne respectent pas cette contrainte.
4. (2 pts) Ajoutez une contrainte dans la relation "usagers" qui garantisse que tout usager inséré a bien au moins un emprunt.
5. (3 pts) Faites deux requêtes qui testent votre contrainte : une requête qui déclenche une exception par le trigger, et une requête qui montre que vous pouvez insérer un nouvel usager (en même temps que son premier emprunt).
6. (bonus 2 pts) Que faudrait-il encore développer pour garantir que notre base de données ne puisse **en aucun cas** avoir un usager qui n'a pas d'emprunt ?