

Travaux Dirigés n°1 : LID - fonctions d'agrégation

EXERCICE 1 : Questions de cours

1. Quels opérateurs du LID SQL peut-on utiliser pour exprimer un maximum ?
2. Quels opérateurs du LID SQL peut-on utiliser pour exprimer un minimum ?
3. Traduire en français la requête suivante :
SELECT ref_produit, designation FROM PRODUITS
WHERE prix_unitaire > ANY (SELECT prix_unitaire FROM PRODUITS);

EXERCICE 2 : Cherchez l'erreur !

On dispose d'une base de données pour la gestion d'un cinéma. Le schéma relationnel est le suivant.

FILMS (no_film, titre, nom_realisateur, nom_producteur)
PROJECTION (no_film#_no_salle, heure, date_proj) - ATTENTION CLE PRIMAIRE
DIFFICILE
ACTEURS (no_acteur, nom, prenom)
JOUER (no_film#, no_acteur#)
SPECTATEURS (no_spect, nom, prenom, age)
VOIR (no_spect#, no_film#)

1. Chacune des requêtes suivantes comporte une erreur. Ecrivez (en français) ce que l'on cherchait à obtenir, et dites où se trouve l'erreur.
 1. SELECT nom, count(no_film) FROM acteurs A, jouer J
where J.no_acteur = A.no_acteur ;
 2. SELECT nom, count(no_film) FROM spectateurs S, voir V
where S.no_spect = V.no_spect GROUP BY S.no_spect;
 3. SELECT nom, count(no_film) FROM spectateurs S, voir V
where S.no_spect = V.no_spect and count(no_film)>=10
GROUP BY S.no_spect, nom;
 4. SELECT nom, count(no_film) FROM spectateurs S, voir V
where S.no_spect = V.no_spect GROUP BY S.no_spect, nom
HAVING (prenom='Jean');
 5. SELECT nom, count(no_film) FROM spectateurs S, voir V
where S.no_spect = V.no_spect GROUP BY S.no_spect, nom
HAVING (no_film>=ALL(select no_film from voir));

EXERCICE 3 :

On s'intéresse maintenant au schéma suivant, qui permet de gérer une bibliothèque.

AUTEUR (code_auteur, nom_auteur, prenom_auteur)
COLLECTION (code_collection, libelle_collection)
GENRE (code_genre, libelle_genre)
LIVRE (no_livre, titre_livre, nom_traducteur, type_public, code_collection#, code_genre#)
ECRIRE (code_auteur#, no_livre#)
LECTEUR (no_lecteur, nom_lecteur, prenom_lecteur, rue_lecteur, ville_lecteur, cp_lecteur, tel_lecteur, date-naissance)
EMPRUNTER (date_emprunt, no_livre #, no_lecteur#, duree)

- 1) Construisez les requêtes suivantes en SQL. L'utilisation de regroupements et/ou de fonctions d'agrégation est nécessaire.
 - a. Nombre total de livres dans la base.
 - b. Nombre de livres par genre de livres (afficher le libellé de chaque genre ainsi que le nombre de livres de ce genre).
 - c. Nombre de livres empruntés par le lecteur no 150.
 - d. Nombre de livres par nom d'auteur.
 - e. Nombre d'auteurs par titre de livre.
 - f. Nombre de lecteurs ayant lu un livre d'Alice Ferney.
 - g. Noms des auteurs qui ont écrit plus de 3 livres.
 - h. Noms des lecteurs ayant emprunté au moins 5 livres, avec le nombre de livres empruntés.
 - i. Durée moyenne d'emprunt par lecteur (pas besoin de faire apparaître le nom du lecteur, mais seulement son code).
 - j. Nombre de livres qui n'ont jamais été empruntés (donnez 3 syntaxes différentes, et parmi ces 3 syntaxes, dites laquelle est spécifique à Postgres).
 - k. Noms du(des) lecteur(s) le(s) plus âgé(s).