Licence d'informatique – 2019/2020

Introduction aux Bases de Données Relationnelles

novembre 2019

Université de Lille 1 SCIENCES ET TECHNOLOGIES

TP6: Base d'une compagnie aérienne

Dans ce TP, nous reprenons la base d'une compagnie aérienne, introduite en TD, pour travailler le TRC. La relation Employees contient des pilotes aussi bien que d'autres métiers. Chaque pilote est certifié pour certains avions, et uniquement les pilotes sont certifiés pour effectuer les vols. Les attributs dep et arr (départ et arrivée) de Vols ont comme domaine soit des chaînes de 3 caractères (les codes AITA des aéroports internationaux), soit simplement les noms des villes. L'attribut portée d'un avion indique la distance maximale qu'il peut parcourir en autonomie. Le schéma est :

Vols(vid :int, dep :varchar(30),arr :varchar(30),distance :int,h_dep :timestamp,h_arr :timestamp,prix :int)
Avions(aid :int,anom :varchar(30),portee :int)
Certifications(eid :int,aid :int)
Employees(eid :int,enom :varchar(30),salaire :int)

Utilisez des vues ainsi que des fonction d'agrégation *bool_and* (alias : EVERY) et *bool_or* quand utile [1]. EVERY sera pour plusieurs questions bien plus facile, que de passer par ∃ et la négation.

Pour les questions 3 et 4, en plus d'écrire les requêtes SQL, vous devrez manipuler des requêtes TRC, c.a.d. les normaliser en utilisant des lois d'équivalence logique.

Question 1 : Pour chaque pilote certifié pour au moins deux avions, donnez l'eid et la portée maximale d'avion pour lesquels ce pilote est certifié.
Question 2 : Déterminez les noms de pilotes dont le salaire est inférieur au prix du vol le moins cher de l'aéroport Paris-Charles de Gaulle (CDG) vers La Tontouta en Nouvelle Calédonie (NOU).
Question 3 : Quelles routes peuvent être volées par tous les pilotes gagnant $>$ 100 000euros ? 2 versions en SQL dont une avec ∃. Puis, en TRC, faites et rendez lors de votre prochain TD sur papier l'exo tel que décrit dans le fichier TRC_queries_student.txt disponible sur Moodle. Vous pouvez le faire electroniquement, et le rendre avec votre TP (mettre en gras au lieu de colorer).
Question 4 : Les noms des pilotes uniquement certifiés pour des avions de portée $>$ 1500 km. Faites deux versions en SQL, l'une avec group by et EVERY, l'autre avec ∃. Faites également la version TRC, voir fichier sur Moodle.
Question 5 : Affichez les noms des pilotes qui sont uniquement certifiés pour des avions avec une portée supérieure à 1500 km, pour au moins deux tels avions.
Question 6 : Donnez les noms des pilotes qui sont certifiés uniquement pour des avions d'une portée supérieure à 1500 km, et qui sont certifiés pour au moins un type de Boeing.
Question 7 : Trouvez l'identifiant de l'employé avec le salaire le *deuxième* plus haut. N'utiliser ni *limit*, ni *order by* !
Question 8 : Affichez les noms des pilotes qui peuvent piloter des avions d'une portée supérieure à 2000km, mais qui ne sont certifiés pour aucun Boeing.

1. https://www.postgresql.org/docs/9.5/static/functions-aggregate.html

Question 9 : Affichez les noms et revenus d'employés qui ne sont pas des pilotes, mais qui gagnent plus que le revenu moyen des pilotes.
Question 10 : Calculez la différence entre le revenu moyen de pilotes, et le revenu moyen de tous les employés (pilotes inclus).
Question 11 : Un client désire voyager de Madison à New York avec au plus deux escales. Donnez la liste des heures de départ à partir de Madison si le client veut arriver à New York avant 18h. [2]
Question 12 : (*** [3]) Est-il possible de prendre une séquence de vols de La Tantouta à Timbuktu ? Chaque vol de la séquence doit partir de la ville qui est l'arrivée du vol précédent. Le premier vol doit partir de La Tantouta, et le dernier vol doit arriver à Timbuktu. Il n'y a aucune contrainte sur le nombre d'escales. Votre requête doit être capable de déterminer si il existe une séquence allant de La Tantouta à Timbuktu, *pour n'importe quelle instance de la relation vols*, donc pas uniquement avec le contenu courant de cette base, mais avec n'importe quel contenu qu'elle pourrait avoir.

2. Aide : testez la fonction postgres $extract(hour from < timestamp >)$.
3. Question bonus. Elle nécessite une technique SQL avancées (vue latérale).