Devoir #1: Partie I — Fondements

Date de remise: Mardi 14 Oct à 6h00 (dans la boite aux devoirs),
Valeur: 10% de la note finale.

Suggestion: Consulter les solutions aux exercices facultatifs, ainsi que les exemples du manuel.

Ce devoir vous donne l'occasion de resoudre plusieurs exercices (légèrement modifiés) du manuel. Ces exercices portent sur le design conceptuel et logique ainsi que sur les requêtes SQL.

1. **(Exercices 2.6 et 3.16) – 50%**. Les membres de SITE voyageant fréquement se sont plaints de la piètre organisation de l'aéroport local d'Ottawa. Suite à ces plaintes, les autorités aéroportuaires ont décidé d'organiser toutes les informations concernant l'aéroport dans une base de données. Votre tâche est de faire le design de cette base de données. La première chose à faire est d'organiser les informations au sujet des avions stationnés et maintenus 1'aéroport. Voici les détails relevants à cette fin:

   - Chaque avion a un numéro d'enregistrement et est d'un modèle donné.
   - L'aéroport acceuille un certain nombre d'avions de divers modèles et chaque modèle est identifié par un numéro de modèle (p.ex. B-747), ainsi qu'une capacité et un poids allant avec ce modèle.
   - Un certain nombre de techniciens travaillent à l'aéroport. Pour chacun d'eux, il faudrait stocker le nom, le NAS, l'adresse, le numéro de téléphone, ainsi que le salaire.
   - Chaque technicien est expert en un ou plusieurs des modèles d'avions et son expertise peut se superposer avec celle d'autre techniciens. Cette information doit aussi être enregistrée.
   - Les contrôleurs aériens ont un examen médical annuel. Stockez la date du plus récent examen pour chacun d'entr'eux.
   - Tous les employés de l'aéroport (techniciens inclus) appartiennent à un syndicat. Stockez cette information pour chacun des employés. Supposez que le NAS identifie exactement chaque employé.
   - L'aéroport conduit un certain nombre de tests périodique pour s'assurer que les avions sont encore en état de voler. Chaque test a un numéro spécifique assigné par l'Administration Fédérale de l'Aviation (FAA), un nom, ainsi qu'un score maximum possible.
   - La FAA requiert que l'aéroport garde des traces de chaque test opéré par un technicien. Pour chacun de ces tests, il faut enregistrer l'information suivante: la date, le nombre d'heures passées à faire ce test et le score obtenu par l'avion au test.

   Sur base des information ci-dessus, faites ce qui suit:

   (a) Dessinez un diagramme ER pour la base de données de l'aéroport. Ce faisant, indiquez les divers attributs de chaque ensemble d'entités et de relations (= associations). Spécifiez les contraintes de clé et participation pour chaque ensemble de relations. Spécifiez aussi toute contrainte de recouvrement nécessaire. Votre diagramme ER ne devrait pas contenir plus de 6 ensembles d'entités et pas plus de 3 ensembles de relations.

   (b) Traduisez votre diagramme ER en un schéma relationnel montrant les instructions SQL nécessaires pour créer les relations du modèle relationnel. N'utiliser que les clés et les contraintes de domaines *null*. Toute contrainte du diagram ER que vous ne pouvez exprimer en SQL doit être expliquée.

   (c) Utilisez une instruction CHECK pour exprimer la contrainte suivante: Seul un technicien expert en un modèle donné peut conduire un test sur ce modèle particulier.

2. **(Exercice 5.8) – 30%**: Considerez les relations suivantes dont la signification est claire:

$Student(\underline{snum} : \texttt{int}, sname : \texttt{string}, major : \texttt{string}, level : \texttt{string}, age : \texttt{int})$
$Class(\underline{cname} : \texttt{string}, meetsat : \texttt{time}, room : \texttt{string}, fid : \texttt{int})$
$Enrolled(\underline{snum} : \texttt{int}, \underline{cname} : \texttt{string})$
$Faculty(\underline{fid} : \texttt{int}, fname : \texttt{string}, deptid : \texttt{int})$

Sur base de ce schéma, formulez les requêtes suivantes **à la fois** en algèbre relationnelle et en calcul relationnel des domaines:

(a) Trouvez les snums des étudiants qui sont inscrits dans exactement deux cours ("classes")

(b) Trouvez les snums des étudiants qui sont inscrits dans tous les cours.

(c) Trouvez les snums des étudiants qui sont inscrits dans tous les cours enseignés par Minsky.

(d) Trouvez les noms des professeurs qui enseignent des cours dans la salle STE0131, mais pas dans la salle STE5201.

(e) Trouvez les noms des professeurs qui enseignent des cours dans la salle STE0131 ou dans la salle STE5201.

Exprimez les contraintes d'intégrité suivantes en SQL, à moins qu'elles soient déjà impliquées par l'une des contraintes de clé primaire ou étrangère du schéma ci-dessus.

(a) Chaque professeur doit enseigner au moins 2 cours.

(b) Deux cours ne peuvent pas avoir lieu au même moment dans la même salle.

(c) Chaque salle de cours ne peut en tout temps contenir qu'une seule classe à la fois.

(d) Les inscriptions à un cours sont limitées à un minimum de 10 étudiants et un maximum de 100.

(e) Chaque étudiant doit être inscrit dans CSI3717.

Finalement, exprimez les requêtes suivantes en SQL:

(a) Pour chaque cours, trouvez le nombre total d'inscriptions pour ce cours.

(b) Trouvez les noms des étudiants qui ne se sont pas inscrits dans CSI5311. (Utilisez une requête imbriquée.)

(c) Trouvez la moyenne d'age des étudiants qui sont agés de plus de 21 ans et cela pour chaque niveau contenant au moins 10 étudiants.

3. **(Exercice 5.10) – 20%**: Considerez les relations suivantes:

$Employee(\underline{eid} : \texttt{int}, ename : \texttt{string}, age : \texttt{int}, sal : \texttt{real})$
$Works(\underline{eid} : \texttt{int}, \underline{did} : \texttt{int}, duration : \texttt{int})$
$Dept(\underline{did} : \texttt{int}, budget : \texttt{real}, managerid : \texttt{int})$

Utilisant ce schéma, formulez des contraintes d'intégrité en SQL (contraintes de domaine, clé, clé étrangère, ou CHECK; ou assertions) ou des triggers SQL qui garantissent que les spécifications (indépendantes) suivantes soient vraies des instances du schéma ci-dessus:

(a) Aucun employé ne peut gagner plus de 1000$.

(b) La durée totale de tout engagement ne peut dépasser 365 (jours).

(c) Chaque fois qu'un employé a une augmentation salariale, le salaire de son manager doit aussi subir une augmentation de maniere à être au moins égal à celui de cet employé au cas où le salaire de ce dernier dépasserait celui du manager.