

BASE DE DONNEES I

(Iluju Kiringa)

Devoir #1: Partie I du manuel —Fondations

Date de remise: Mardi 24 février, 2015 (en classe)

Valeur: 5%

SUGGESTION: Inspirez vous des exemples du manuel!

Les exercices dans ce devoir sont issus de notre manuel. Parfois, je modifie ceux du manuel pour les rendre un peu plus difficile ou plus facile selon les cas. Le thème de ces exercices est principalement le design conceptuel et logique de base de données et requêtes SQL.

1. **(Exercices 2.6 et 3.16) – 50%.** Supposez que l'information relative à l'aéroport d'Ottawa est organisée dans une base de données et que vous êtes appelé à concevoir la dite base de données. Le prérequis suivants concernent l'organisation de l'information sur les avions stationnés et entretenus à l'aéroport :

- Chaque avion a un numéro d'enregistrement et est d'un modèle spécifique.
- L'aéroport accueille un certain nombre de modèles d'avion, et chaque modèle est identifié par un numéro de modèle (par exemple B-747), ainsi qu'une capacité et un poids.
- Un certain nombre de techniciens travaille à l'aéroport. Pour chacun d'eux, conserver son nom, NAS, adresse, numéro de téléphone et salaire.
- Chaque technicien est un expert sur un ou plusieurs modèles d'avion, et son expertise peut-être se chevaucher avec celle d'autres techniciens. Cette information sur les techniciens doit aussi être enregistrée.
- Les contrôleurs aériens ont un examen médical annuel. Pour chacun d'eux, enregistrez la date du plus récent examen.
- Tous les employés de l'aéroport (y compris les techniciens) appartenant à un syndicat. L'affiliation syndicale de chaque employé sera stockée. Vous pouvez supposer que chaque employé est identifié de manière unique par un NAS.
- L'aéroport a un certain nombre de tests qui sont utilisés régulièrement pour s'assurer que les avions sont encore en état de navigabilité. Chaque test a un numéro de test FAA, un nom et un score maximal possible.
- La FAA exige que l'aéroport garde une trace de chaque fois qu'un avion donné est testé par un technicien donné à l'aide d'un test donné. Pour chaque test, les informations nécessaires sont la date du test, le nombre d'heures que technicien a passé à faire le test et le score que l'avion a reçu sur le test.

Selon les informations données ci-dessus, procédez comme suit:

- (a) Dessiner un diagramme ER de la base de données de l'aéroport. Ce faisant, indiquez les différents attributs des ensemble d'entités et des ensembles de relations ; Spécifiez également les contraintes de clé et de participation pour chaque ensemble de relations. Spécifier des contraintes de chevauchement nécessaires. Votre diagramme ER ne doit pas contenir plus de 6 ensembles d'entités et pas plus de 3 ensembles de relations.
 - (b) Traduire votre diagramme en un schéma de base de données relationnelle en montrant les instructions SQL requises pour créer les relations, en faisant usage de contraintes clés primaires et étrangères, ainsi que des contraintes de *null*. Toute contrainte du diagramme que vous ne pouvez pas capturer dans vos instructions SQL doit être documentée.
 - (c) Supposons la contrainte supplémentaire que seulement un technicien qui est un expert sur un modèle d'avion peut effectuer un test sur ce modèle particulier. Exprimez cela en utilisant l'instruction CHECK.
2. **(Exercice 5.8) – 30%.** Examinez les relations suivantes dont la signification est évidente:

***Students*(snum: int, sname: string, major: string, level: string, age: int)**

***Class*(cname: string, meetsat: time, room: string, fid: int)**

***Enrolled*(snum: int, cname: string)**

***Faculty*(fid: int, fname: string, deptid: int)**

Selon ce schéma, formulez les requêtes suivantes, tant en algèbre relationnelle qu'en calcul relationnel (domaine):

- (a) Trouver les *snums* des étudiants qui sont inscrits dans exactement deux cours (classes).
- (b) Trouver les *snums* des étudiants qui suivent tous les cours.
- (c) Trouver les *snums* des étudiants qui suivent tous les cours donnés par Minsky.
- (d) Rechercher les noms des membres du corps professoral qui enseignent dans la salle STE0131, mais pas dans la salle STE5201.
- (e) Rechercher les noms des membres du corps professoral qui enseignent dans les salles s STE0131 ou STE5201.

En outre, exprimez les contraintes d'intégrité suivantes en SQL, sauf si elles sont impliquées par les contraintes de clés primaires et étrangères qui peuvent être exprimées sur le schéma ci-dessus :

- (a) Chaque membre du corps professoral doit enseigner au moins deux cours.
- (b) Deux cours ne peuvent pas avoir lieu dans la même pièce en même temps.
- (c) Chaque salle ne peut accueillir qu'un seul cours à la fois.
- (d) Chaque cours a un effectif d'au moins 10 étudiants et au maximum de 100.
- (e) Tous les étudiants doivent être inscrits à CSI2532.

Enfin, exprimez les requêtes suivantes dans SQL :

- (a) Pour chaque cours, trouver le nombre total d'inscriptions pour ce cours.
- (b) Rechercher les noms des étudiants qui ne se sont pas inscrits dans csi5311. (Utilisez une requête imbriquée)
- (c) Trouver l'âge moyen des étudiants qui ont plus de 21 ans pour chaque niveau qui contient au moins 10 étudiants.

3. **(Exercice 5.10) – 20%.** Examinez les relations suivantes dont la signification est évidente:

***Employee*(eid: int, ename: string, age: int, sal: real)**

***Works*(eid: int, did: int, duration: int)**

***Department*(did: int, budget: real, managerid: int)**

En vous basant sur ce schéma, écrivez des contraintes d'intégrité en SQL (domaine, clé, clé étrangère, ou CHECK / ASSERTION) pour vous assurer que les exigences suivantes sont satisfaites par les instances du schéma ci-dessus :

- (a) Aucun salarié ne peut avoir plus de 1000\$ de salaire.
- (b) La durée totale de toutes les nominations d'un employé ne doit pas dépasser 365 jours.
- (c) Chaque fois qu'un employé reçoit une augmentation de salaire, le salaire de son gestionnaire (manager) doit augmenter pour être au moins autant que celui de cet employé, dans le cas où le salaire dudit employé serait plus grand ou égal au salaire du gestionnaire.