Examen final

Master 1 Ingénierie Mathématique

Bases de données relationnelles

Benjamin Auder

26 mars 2014

Exercice 0 [Environ 0% des points]

PostgreSQL et PHP ont un point commun de taille. Quel est-il ?

Exercice 1 [Environ 25% des points]

Cet exercice a pour objectif la construction d'une base de données modélisant l'organisation d'une piscine, selon la description donnée ci-après.

— Il y plusieurs bassins de tailles et profondeurs différentes.
— Un bassin est ouvert au public et/ou pour des cours pendant certains créneaux horaires ; les horaires d'ouvertures varient au fil de la semaine.
— Un bassin ouvert est toujours surveillé par un maître-nageur.
— Un maître-nageur peut donner des cours, éventuellement à thème.
— Un cours − hebdomadaire − est donné dans un certain bassin à un certain créneau.
— Un nageur paye son entrée à un tarif variant en fonction de sa situation professionnelle, et pouvant être réduit par un abonnement de 10 séances.
— Un nageur participant à un cours thématique est susceptible d'avoir un record enregistré (par exemple 5min05 au 400m 4 nages).

Exemples de requêtes devant pouvoir être effectuées sur la base :
— nombre de nageurs inscrits au cours donné par Nathalie le mercredi à 18h ;
— nom du surveillant du bassin de 50m le dimanche matin ;
— dépenses totales pour les cours du nageur numéro 32.

(a) Dessinez et commentez un schéma entités/associations correspondant à cette situation. Ensuite, normalisez le schéma comme indiqué au second cours vers le slide 20 ; les étapes ne sont pas toutes cruciales ou applicables, mais vous devriez au moins :
— utiliser des noms communs pour les entités et leurs attributs, et des verbes (éventuellement au passif et/ou avec une préposition) pour les associations ;
— souligner les attributs identifiants ;
— indiquer les cardinalités.
Le schéma peut être dessiné sur la copie d'examen, ou conçu via un logiciel.
Suggestion 1 : http://code.google.com/p/merisier/wiki/Usage. Il est installé sur la machine 192.168.31.236, mais pas complètement au point : les diagrammes nécessitent quelques ajustements manuels.
Suggestion 2 : https://www.draw.io/# (moins spécialisé ; choisir "entity relationship" dans le menu à gauche).

(b) Comment étendre la modélisation pour tenir compte d'éventuelles compétitions ? Celles-ci se définissent par leurs lieu et date, et on veut mémoriser les résultats des membres du club ; avec les éventuelles médailles le cas échéant. On ne prendra pas en compte cette extension pour la question suivante.

(c) Indiquez et justifiez les transformations à effectuer pour obtenir un schéma relationnel. Vous pouvez éventuellement détailler certaines parties du schéma si cela semble

pertinent. Expliquez ensuite ensuite comment obtenir une instruction CREATE TABLE à partir d'une table dans le schéma relationnel. Appliquez cet "algorithme" de transformation à une table qui en référence d'autres.

Exercice 2 [Environ 25% des points]

On considère les schémas de tables ci-dessous, représentant des pages web interconnectées appartenant à des sites. Les attributs auteur dans Page et webmaster dans Site référencent l'identifiant de la relation "rédacteurs". Les attributs page1 et page2 dans la relation Lien référencent l'attribut chemin dans une Page, et site1 site2 sont les URLs des sites de départ et d'arrivée.

L'url (complète) d'une page est constituée de celle du site auquel elle appartient, et d'un suffixe indiquant le chemin vers la page à partir du site (l'attribut "chemin" dans la relation Page). Par exemple url du site = "http ://tex.stackexchange.com" et chemin vers la page = "/questions/13173/how-to-change-the-position-of-underset". Enfin, il peut coexister différentes versions d'une page à des dates différentes (dans un wiki par exemple) ; c'est pourquoi date_màj fait partie de la clé de la relation Page.

— Rédacteur (id, nom, email)
— Site (url, #webmaster, thème)
— Page (#site, chemin, date_màj, #auteur, contenu)
— Lien (#site1, #page1, #site2, #page2)

Remarque : le modèle relationnel n'est pas idéal pour représenter un graphe. Une recherche google avec "graph database" conduit à d'autres types de SGBD plus adaptés.

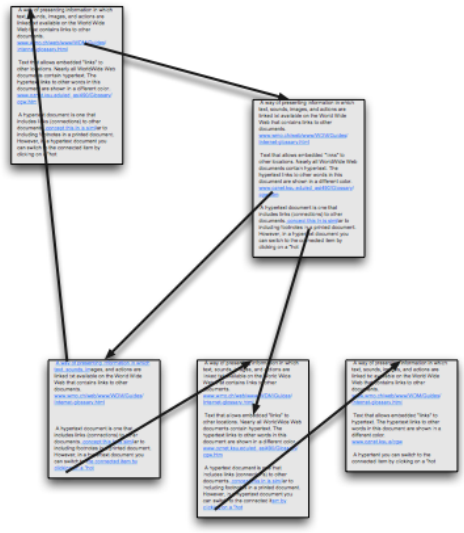

FIGURE 5 – Illustration

Écrivez les requêtes suivantes en SQL. Exprimez les trois premières également en algèbre relationnelle (AR). Si vous choisissez le format informatique vous pouvez écrire “sél[ection]”, “proj[ection]”, “join[ture]” ...etc au lieu de rechercher les symboles ; du moment qu'il n'y a pas d'ambiguité ça me va.

(a) Les chemins vers les pages rédigées par l'auteur d'identifiant 32 dans le site ayant pour url www.starwars.com

(b) Les noms des auteurs ayant mis à jour une page en 2014.

(c) Les URLs des sites comportant des liens vers une page de www.disney.fr

(d)
   i. Les sites ayant au moins un lien vers une page rédigée par Luke,
   ii. rangés par nombre total de liens (ne vérifiant pas nécessairement la condition) décroissant. Vous pouvez réutiliser la requête précédente avec WITH.

(e) Toutes les URLs pouvant être atteintes à partir du site www.disney.fr,
   i. en une étape, rangées par nom de domaine puis par chemin ;
   ii. en un nombre quelconque d'étapes ;

   en excluant les pages du site de départ dans les résultats de recherche.

(f) la dernière activité connue des auteurs des pages du site www.nintendo.fr (pas forcément sur ce même site). Retournez les URLs des pages, les noms des auteurs et les dates de mise à jour.

(g) Les auteurs d'une page sur le thème “annonces” contenant un numéro de téléphone, qui sont aussi webmasters d'un site sur un thème différent.

Toutes les requêtes SQL peuvent être testées en se connectant à la base “web” présente sur 192.168.31.236. Vous disposez de deux alternatives pour les tests.

Connexion sur la machine des TPs :

1. ouvrir un terminal
2. ssh initiales@192.168.31.236 #mot de passe par défaut : tpsql
3. psql #mot de passe par défaut : tpsql

Passage par l'interface web :

1. récupération de votre mot de passe si vous ne le connaissez pas :
   ssh initiales@192.168.31.236 #mot de passe par défaut : tpsql
   cat .web_password
2. naviguez sur http ://192.168.31.236
   login = vos initiales
   mot de passe = la sortie de cat .web_password ci-dessus.
3. choisissez “TP2 : écrire des requêtes SQL” puis la base de données “web”

Exercice 3 [Environ 25% des points]

On considère dans cet exercice la relation Cours (C, P, H, S, E, N) dont les attributs représentent respectivement un cours, un professeur, une heure, une salle, un étudiant et une note. Les dépendances fonctionnelles de cette relation sont

$$\begin{aligned} C &\to P \\ HS &\to C \\ HP &\to S \\ HE &\to S \\ CE &\to N \end{aligned}$$

(a) Sans effectuer de calculs, juste en observant les dépendances fonctionnelles, déterminez le cardinal minimal d'une clé de la relation Cours. Justifiez votre réponse.

(b) Cherchez alors une clé minimale pour la relation Cours. Utilisez-la pour les questions suivantes.

(c) Y a-t-il des dépendances fonctionnelles redondantes ? (C'est-à-dire des DF pouvant être déduites à partir d'autres par application des axiomes d'Amstrong ; parcourez éventuellement cette lecture complémentaire intéressante, mais ne vous y perdez pas, vous n'avez pas besoin de tout ça). Si oui, continuez l'exercice sans elles.

(d) Quelles sont les formes normales vérifiée par la relation Cours ? Justifiez la réponse.

(e) Décomposez la relation Cours en un ensemble de relations en 3NF en utilisant les théorèmes du paragraphe 3.3.2. Ces relations sont-elles aussi en BCNF ?

(f) Interprétez la décomposition obtenue : vous semble-t-elle cohérente par rapport à la sémantique des attributs ? Que remarquez-vous ?

(g) La décomposition est-elle sans perte de dépendances ? Pouvez-vous la transformer intuitivement en une décomposition vérifiant ce critère ? (Un algorithme général est présenté ici, mais vous n'en avez pas vraiment besoin).

Exercice 4 [Environ 25% des points]

On considère les schémas de relations suivants, modélisant les commandes passées dans un restaurant. Celles-ci peuvent être consommées sur place ou emportées ; dans ce dernier cas on effectue une remise de 10% sur le prix total. Certaines parties de la commande peuvent être offertes (les apéritifs par exemple). Les attributs commande et produit référencent respectivement l'identifiant d'une commande et d'un produit.

— Produit (<u>id</u>, type, nom, prix_unitaire, stock)
— Entête_commande (<u>id</u>, prix, à_emporter, date)
— Détail_commande (#<u>commande</u>, #<u>produit</u>, quantité, offert)

Les commandes SQL générant les tables correspondantes sont accessibles en lecture dans le répertoire /home/tpbd/m1im sur la machine 192.168.31.236. Chargez-les dans votre base en tapant "\i /home/tpbd/m1im/restaurant.sql" une fois dans psql ou "psql -f /home/tpbd/m1im/restaurant.sql" depuis un terminal. Cela permettra de tester vos réponses au fur et à mesure.

Rappel :

L'interaction avec PostgreSQL est possible par connexion sur la machine des TPs :

1. ouvrir un terminal
2. ssh initiales@192.168.31.236
3. psql -d web

Ou bien en passant par l'interface web :

1. naviguez sur http ://192.168.31.236
2. choisissez "TP3 : connexion à votre base"

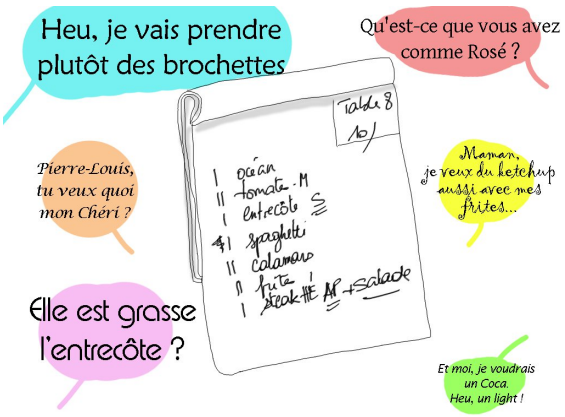

FIGURE 6 – Illustration

Implémentez les fonctions suivantes.

(a) Trigger verifieMajDetail() qui vérifie les disponibilités et met à jour les stocks lors de l'ajout, modification ou suppression d'un détail de commande.

(b) Fonction finaliseCommande() prenant en argument un identifiant de commande : enregistre la date, calcule le prix total et le met à jour. Cherchez aussi un moyen d'empêcher toute future modification ou suppression de la commande. Vous aurez probablement besoin d'un trigger.

(c) Fonction imprimeAddition() prenant en argumant un identifiant de commande, et retournant une table correspondant à l'addition : trois colonnes, "quantité", "nom du produit" et "prix". La dernière ligne ne contient rien dans la première colonne, puis "Total" dans la seconde et le prix total (supposé calculé) dans la troisième.