

DS d'informatique n°2 : corrigé

PTSI Lycée Eiffel

11 mars 2016

Exercice 1

1. On peut raisonnablement supposer que Matricule sera une clé primaire possible pour l'entité Chirurgien, et Numéro pour l'entité Bloc opératoire. Pour l'entité Opération, il faut manifestement rajouter un attribut du genre n°opération.
2. L'entité Opération est reliée aux deux autres, avec des cardinalités 1..n partout sauf dans le sens Opération \rightarrow Bloc Opératoire où on aura une cardinalité 1..1 (un seul bloc pour une opération).
3. Le lien entre Opération et Bloc Opératoire étant de type 1..n 1..1, pas besoin de créer de nouvelle table, il suffira de mettre dans la table Opération un attribut n°bloc correspondant au numéro du bloc opératoire où a lieu l'opération. Par contre, le lien entre Opération et Chirurgien est de type 1..n 1..n, il faut créer une nouvelle table Participation ayant simplement deux attributs : Matricule et n°opération, permettant de faire la liste des opérations auxquelles ont participé chaque chirurgien. On aura donc les quatre tables suivantes (la précision des types de données est facultative) :
 - **Chirurgien** (Matricule VARCHAR(10), Nom VARCHAR(20), Prénom VARCHAR(20), Adresse mail VARCHAR(40), expérience INT)
 - **Bloc Opératoire** (Numéro INT, Étage INT, Type VARCHAR(20))
 - **Opération** (n°opération INT, Type VARCHAR(20), Date DATE, Heure début INT, Heure fin INT, n°bloc INT)
 - **Participation** (Matricule VARCHAR(10), n°opération INT)
4. CREATE TABLE Bloc Opératoire (Numéro INT, Étage INT, Type VARCHAR(20) DEFAULT 'Chirurgie générale')
5. (a) $\pi_{\text{Nom, Prénom}}(\sigma_{\text{expérience} \geq 5}(\text{Chirurgien}))$
(b) $\pi_{\text{Date}}(\sigma_{\text{Heure début} < 8}(\text{Opération}))$
(c) $\pi_{\text{Type}}(\sigma_{\text{Nom} = \text{'Denis Charcute'}}((\text{Chirurgien} \bowtie_{\text{Chirurgien.Matricule} = \text{Participation.Matricule}} \text{Participation}) \bowtie_{\text{Participation.n°opération} = \text{Opération.n°opération}} \text{Opération}))$
6. (a) SELECT n°opération FROM Opération WHERE (Date fin - Date début) > 3
(b) SELECT Nom, Prénom FROM Chirurgien C, Participation P, Opération O, Bloc Opératoire B WHERE C.Matricule=P.Matricule AND P.n°opération=O.n°opération AND n°bloc=B.Numéro AND B.Type='Chirurgie dentaire'
(c) SELECT Bloc Opératoire.Numéro FROM Bloc Opératoire B, Opération O WHERE B.Numéro=n°bloc AND Date='2016-03-11'

Exercice 2

1. Ce sont des clés secondaires correspondant aux clés primaires n°client (pour l'attribut client) et n°id (pour pizza) des tables Pizzas et Clients. Bien sûr, il aurait été plus clair de leur donner le même nom que dans leur table d'origine, mais en pratique, on donne souvent aux

clés secondaires le nom de leur table d'origine plutôt qu'un nom peu informatif du type n°id. On ne peut pas vraiment prendre le couple (client,pizza) comme clé primaire de la table Commande, car cela supposerait qu'un même client ne peut pas commander une pizza donnée plusieurs fois (à des dates différentes), ce qui n'est pas raisonnable.

2. On choisira certainement le type INT pour n°commande, ainsi que pour client et pizza puisqu'on a considéré qu'ils correspondaient à n°client et n°id. Pour date, on peut bien sûr utiliser le type SQL Date, ou plus simplement un VARCHAR(20).
3. (a) On affiche le nom de toutes les pizzas, avec leur prix.
 (b) On affiche le nom et le prénom de tous les clients habitant une ville dont le nom commence par B.
 (c) On affiche le nom et l'adresse de tous les clients ayant effectué une commande le 5 février 2015.
 (d) On affiche le nom et le prénom de tous les clients ayant déjà commandé une pizza Quatre fromages.
4. Le plus simple a priori est d'ajouter dans la relation commande un attribut Quantité, indiquant le nombre de pizzas (de la catégorie correspondant au n°id référencé par l'attribut pizza) contenus dans cette commande. Mais ça ne répond pas exactement à ce que demande l'énoncé, puisque dans ce cas, une commande constituée de plusieurs types de pizzas (par exemple trois Hawaïennes et deux Calzone) sera « découpée » en deux lignes (et donc deux n°commande différents) dans la table Commande (on ne peut pas mettre deux pizzas différentes sur une même ligne). Une meilleure solution consiste en fait à supprimer l'attribut pizza de la table Commande, et à créer une nouvelle table **PizzaCommande** contenant les attributs n°commande, pizza, quantité (le couple formé des deux premiers constituant une clé primaire de la table), et permettant d'associer à un même numéro de commande plusieurs types de pizzas avec les quantités correspondantes. C'est cette solution qu'on supposera adoptée pour la fin de l'exercice.
5. (a) `SELECT Nom, Prénom FROM Clients, Commande, PizzaCommande WHERE client=n°client AND Commande.n°commande=PizzaCommande.n°commande AND pizza='Calzone' and quantité=10`
 (b) `SELECT COUNT(PizzaCommande.n°commande) FROM Clients, Commande, PizzaCommande WHERE clien=n°client AND Commande.n°commande=PizzaCommande.n°commande AND Nom='Duchmol' AND Prénom='Cyril' AND pizza='Hawaïenne' AND date>='2016-01-01'`
 (c) Cette dernière question est un gros piège, car les restrictions de syntaxe SQL rendent le calcul assez pénible. En fait, on aimerait naturellement créer une table contenant chaque n°id de pizza avec dans une deuxième colonne le nombre total de commandes concernant cette pizza, et en sortir le maximum. Une syntaxe « naturelle » serait quelque chose du genre `SELECT n°id, MAX(SUM(quantité)) From PizzaCommande`, mais ça n'est pas du tout autorisé en SQL ! Il faut en fait combiner la fonction d'agrégation (SUM, MAX ou une autre) avec un GROUP BY permettant de regrouper toutes les lignes d'un tableau ayant la même valeur pour un certain attribut. Ainsi, `SELECT n°id, SUM(quantité) FROM PizzaCommande GROUP BY n°id` va faire un tableau contenant exactement ce qu'on veut. On peut bien sûr rajouter des conditions avec un WHERE à condition de les mettre avant le GROUP BY. Une solution pour ce qui était demandé est alors la suivante : `SELECT Nom FROM Pizza, Commande, CommandePizza WHERE n°id=pizza AND Commande.n°commande=CommandePizza.n°commande AND date>='2016-01-01' AND n°id = (SELECT n°id FROM (SELECT n°id, SUM(quantité) FROM PizzaCommande GROUP BY n°id)`