

TD2 & TP2 : la Grande Bouffe

1 Position du problème et schéma des relations

Une maîtresse de maison a constitué une base de données relationnelle sur les *personnes* qu'elle invite. Elle représente, pour chaque *repas* identifié par une date, l'ensemble des *plats* qui ont été servis et pour chaque plat le *vin* qui l'accompagnait. Pour les *amis*, pas forcément invités, elle connaît leurs *plats préférés*. Pour chaque plat (servi ou apprécié) elle associe son type.

Cette maîtresse de maison fort avisée définit le schéma de relations suivant (les identifiants des relations sont les attributs soulignés) :

LesRepas (dateR, nomI)

/ (d, i) ∈ LesRepas ⇔ la personne de nom i, a été invitée au repas identifié par la date d */*

LeMenu (dateR, nomP, nomV)

/ (d, p, v) ∈ LeMenu ⇔ le plat p accompagné par le vin v a été servi au repas d */*

LesPréférences (nomA, nomP)

/ (n, p) ∈ LesPréférences ⇔ l'ami de nom n aime le plat de nom p */*

LesPlats (nomP, typeP)

/ (p, t) ∈ LesPlats ⇔ le plat p est du type t */*

domaine (dateR) = date

domaine (nomI) = domaine (nomP) = domaine (nomV) =

domaine (typeP) = domaine (nomA) = chaînes de caractères

2 Compréhension et lecture

Question 1 :

Compléter le schéma donné dans l'énoncé par l'expression des contraintes d'intégrité référentielle. Illustrer les contraintes d'intégrité référentielle via des valeurs dans les relations (au moins deux dates différentes, quatre plats, deux personnes invités et trois amis avec ses préférences)

Question 2 :

Les expressions suivantes sont-elles correctes ? Si non, pourquoi ? Si oui, pour chacune, donnez la spécification de la relation qu'elle construit (attribut, identifiant, prédicat).

1. SELECT nomI
FROM Plats
WHERE nomP = 'mousse';
2. SELECT DISTINCT nomI
FROM LesRepas
WHERE TO_CHAR(dateR, 'DD-MM-YYYY') = '31-12-2017';
3. SELECT *
FROM LesRepas R JOIN LeMenu M ON (R.dateR=M.dateR)
WHERE nomP = 'Ile flottante';
4. SELECT *
FROM LesRepas JOIN LeMenu USING (dateR)
WHERE nomP = 'Ile flottante';

2.1 Expression de requêtes

Question 3 :

Pour chaque requête ci-dessous, spécifier la relation construite par le résultat attendu, et donner son expression en SQL. Les requêtes devront construire des résultats sans répétition de valeurs. La clause **DISTINCT** sera utilisée uniquement lorsque nécessaire. Les produits de relation seront exprimés dans la clause **FROM** des requêtes.

1. Donner l'ensemble des vins qui ont été servis avec un médaillon de langouste.
2. Donner les noms des plats (avec la boisson qui les accompagne et la date) servis le 21 octobre 2016
3. Donner les noms des invités qui ont mangé au moins une fois du foie gras.
4. Donner les noms des amis qui n'ont jamais été invités
5. Donner les repas (date, plats et vins servis) auxquels Marie et Adrian ont été invités (ensemble).
6. Donner la liste des amis qui ont eu, au moins une fois, un plat de leurs préférences.
7. Donner les noms des amis qui aiment seulement les desserts.
8. Donner les noms des personnes qui ont été invitées au moins 2 fois.

Indication : ne pas utiliser les opérateurs de partition et d'agrégation de SQL.

9. Donner les noms des personnes qui ont été invitées exactement 2 fois.

Indication : ne pas utiliser les opérateurs de partition et d'agrégation de SQL.

10. Donner l'ensemble des amis qui n'ont jamais eu un de leurs plats préférés.
11. Donner les noms des amis invités qui ont mangé du Foie gras, mais qui n'aiment pas cela.