

TD4 & TP4 : Agence de voyages

1 Position du problème

Une agence de voyage a informatisé la gestion des voyages qu'elle propose (itinéraires, monuments visités, réservations, etc.). La base de données a été construite à partir du cahier des charges suivant :

Les circuits :

Un circuit est identifié par un numéro, il est décrit par une ville de départ, une ville d'arrivée et une séquence d'étapes. Une étape se déroule pendant un nombre donné de jours, dans une ville donnée. Au cours de chaque étape, tous les monuments de la ville, lorsqu'il y en a, sont visités. Les villes de départ et d'arrivée n'étant pas considérées comme des étapes, leurs monuments ne sont pas visités.

Un même circuit ne repasse jamais plusieurs fois dans la même ville étape, mais il peut arriver qu'une ville départ ou arrivée figure aussi dans l'ensemble des villes étapes. Ceci permet de prendre en compte les situations où les villes de départ et/ou d'arrivée font l'objet d'une visite. Les villes sont identifiées par leur nom. Les monuments sont identifiés par leur nom, dans la ville où ils sont situés.

Un circuit peut être programmé plusieurs fois, à des dates différentes. A chacune de ces programmations, on associe un nombre de places proposées (c'est-à-dire le nombre maximal de places qu'il est possible de réserver). Deux programmations d'un même circuit peuvent avoir des nombres différents de places proposées. Par contre, le prix d'un circuit est fixé, toujours le même quelque soit sa programmation. Un circuit dure un nombre de jours égal à la somme des durées de chacune de ses étapes.

Les réservations :

Une réservation, identifiée par un numéro, est effectuée pour le compte d'un client (identifié par son nom) et concerne une programmation d'un circuit. Plusieurs places pour la même programmation du même circuit peuvent être réservées en une seule fois.

Pour chaque programmation, le nombre total de places réservées calculé sur toutes les réservations de cette programmation est inférieur ou égal au nombre de places offertes de la programmation.

2 Schéma des relations

Le schéma de la base de données est donné ci-dessous (les identifiants sont soulignés) :

LesVilles (nomV, pays)

/ (n, p) ∈ LesVilles ⇔ la ville dont le nom est n, est située dans le pays p. Le nom de la ville est un identifiant (clef de la relation). */*

LesMonuments (nomM, nomV, prix)

/ (nm, nv, p) ∈ LesMonuments ⇔ le monument de nom nm est situé dans la ville nv. Son prix de visite est p euros. */*

LesCircuits (numC, vDep, vArr, prix)

/ (n, vd, va, pr) ∈ LesCircuits ⇔ le circuit touristique identifié par le numéro n, part de la ville vd et se termine dans la ville va. Son prix est de pr, qui ne prend pas en compte le prix des monuments visités. La ville de départ représente le point de rendez-vous avec les accompagnateurs. */*

LesEtapes (numC, rang, vÉtape, nbJours)

/ (n, r, ve, nbj) ∈ LesEtapes ⇔ la r^e étape du circuit n se déroule dans la ville ve, où le séjour est de nbj jours. On fait comme hypothèse que lorsqu'une ville est dans un circuit, tous ses monuments sont visités. r ≥ 1. Les villes de départ et d'arrivée (vArr et vDép de LesCircuits) sont dans LesEtapes lorsqu'elles sont visitées. */*

LesProgrammations (numC, dateDep, nbPlaces)

/ (n, d, nbl) ∈ LesProgrammations ⇔ le circuit identifié par le numéro n, programmé à la date d est offert avec nbl places. Le même circuit peut être programmé à différentes dates. */*

LesRéservations (numR, nomC, numC, dateDep, nbRes)

/ (nr, no, nc, d, nbr) ∈ LesRéservations ⇔ le client de nom no, a effectué une réservation identifiée par nr, sur le circuit nc programmé à la date d. Il a réservé nbr places. */*

Les domaines associés sont :

domaine (nomC) = {'Bonemine', 'Corto', etc.}
 domaine (vEtape) = domaine (vDep) = domaine (vArr)
 = domaine (nomV) = {'Paris', 'Florence', 'Briançon', etc.}
 domaine (pays) = {'Italie', 'Finlande', 'France', etc.}
 domaine (numC) = domaine (numR) = domaine (nbRes) = domaine (rang)
 = domaine (nbJours) = domaine (prix) = domaine (nbPlaces) = entiers > 0
 domaine (dateDep) = instant (unité jour)

Les contraintes d'intégrité référentielle sont :

LesReservations[numC, dateDep] \subseteq LesProgrammations[numC, dateDep]
 LesProgrammations[numC] = LesCircuits[numC]
 LesEtapas[numC] \subseteq LesCircuits[numC]
 LesEtapas[vEtape] \subseteq LesVilles[nomV]
 LesCircuits[vDep] \subseteq LesVilles[nomV]
 LesCircuits[vArr] \subseteq LesVilles[nomV]
 LesMonuments[nomV] \subseteq LesVilles[nomV]

Remarques :

- On dit qu'un circuit passe par une ville v, lorsque v est une de ses étapes, ou/et la ville arrivée, ou/et la ville de départ ;
- On dit qu'un circuit visite une ville lorsque celle-ci est une étape de ce circuit.
- On généralise ces définitions aux pays : un circuit passe dans un pays p lorsque l'une de ses étapes, et/ou sa ville arrivée, ou/et sa ville de départ sont situées dans p ; un circuit visite un pays lorsque celui-ci a une fois au moins une étape dans ce pays.

3 Expressions de requêtes

- Les requêtes devront construire des résultats ordonnés et sans répétition de valeurs, la clause **DISTINCT** ne sera utilisée que lorsque nécessaire.
 - Les sous-requêtes dans le **FROM** seront exprimées dans la clause **WITH AS**.
1. *A faire en TP : Fournir un fichier SQL (extension .sql) avec des commentaires (-) détaillant les tests réalisés (requêtes auxiliaires) pour vérifier que vos requêtes sont correctes.*
 2. Donner les noms des clients qui ne visitent aucun monument. Schéma attendu (nomC)
 3. Donner les noms des monuments (avec les villes concernées) visités par Milou. Schéma attendu (nomM, vEtape)
 4. Donner les noms des villes visitées par Milou avec le nombre de monuments dans chaque ville (faire apparaître aussi les villes sans monument avec un 0). Schéma attendu (vEtape, nbMon)
 5. Donner le numéro, les villes de départ et d'arrivée des circuits qui démarrent après le 20 Décembre 2010. Schéma attendu (numC, vDep, vArr)
 6. Donner le(s) circuit(s) qui a(ont) le plus grand nombre d'étapes et donner aussi sa(leur) durée. Schéma attendu (numC, nbJoursTot)