

Cahier des charges : Achat de séjours de vacances par des clients

Une agence de voyages propose à ses clients des séjours dans des villages de vacances, qu'ils peuvent choisir, et payer grâce à un avoir.

Données La base suivante décrit les informations correspondantes.

Client(idc, nom, age, avoir) Village(idv, ville, activite, prix, capacite) Sejour(ids, idc, idv, jour)

Un client a un identifiant, un nom, un âge, et un avoir (entier).

Un village de vacances a un identifiant, une ville, l'activité qu'on peut y pratiquer, un prix par jour (entier) et un nombre de lits (capacité).

Un séjour a un identifiant, un identifiant d'acheteur, un identifiant de village, et une date. Pour simplifier, chaque séjour a une durée d'un jour, et on considère que tout à lieu dans l'année courante, une date sera donc simplement un jour : un entier entre 1 et 365.

Exemple *Riton* a 22 ans, et un avoir de 1700 euros. Il y a un village à *Rio*, de 250 lits, coût par jour 50 euros, où l'on pratique le *kitesurf*. Le client 2 a acheté un séjour dans le centre 11 pour le 31 décembre.

Client(1, Riton, 23, 1700) Village(10, Rio, kitesurf, 50, 250) Sejour(100, 2, 11, 365)

Précisions et restrictions Les identifiants (entiers) sont uniques. Plusieurs clients peuvent avoir le même nom. L'agence fait un cadeau de bienvenue aux nouveaux clients, qui, en s'inscrivant, se voient créer automatiquement un avoir de 2000 euros, mais ensuite les avoirs négatifs ne sont pas autorisés. Plusieurs villages peuvent être dans la même ville. Il n'y a qu'une activité par village. Un client ne peut acheter un séjour que pour lui-même, donc un seul pour une date donnée. Il peut y avoir d'autres objets SGBD dans la base (tables, contraintes, etc.), mais on ne les connaît pas.

Accès aux données Les employés ont des comptes SGBD sur la base, et il y a un unique compte client. Un employé peut créer des villages, les consulter et les modifier (sauf identifiant, ville et prix), consulter les séjours, et effectuer le traitement 3. Un client peut effectuer les traitements 1 et 2, consulter les villages pour lesquels il n'y a aucun séjour (mais sans la capacité), ainsi que consulter les lignes lui correspondant dans chaque table à partir de son identifiant (et pour *Client* il donne aussi son nom). Rien d'autre n'est possible.

Fonctionnement Les traitements ne vérifieront pas les contraintes, ni si les paramètres sont corrects.

Traitement 1 : L'inscription d'un client est faite par lui-même comme suit. Il donne son nom et son âge, une nouvelle ligne est alors créée dans *Client*, et il obtient son identifiant. Paramètre(s) : nom et âge. Retour(s) : identifiant client.

Traitement 2 : L'achat d'un séjour est fait par un client comme suit. Il arrive avec en tête une ville et un jour. Un village dans cette ville de prix journalier le plus cher est alors cherché (car plus le prix est élevé, meilleur est le village, et l'activité n'est pas prioritaire). S'il y en a plusieurs l'un quelconque est choisi (par exemple le premier qui vient). Puis le séjour correspondant est créé et l'avoir du client décrémente du prix journalier du village. S'il n'y en a aucun on ne fait rien. Paramètre(s) : identifiant client, ville et jour. Retour(s) : identifiants village et séjour, et activité ; sinon -1, -1, *néant* si pas de village dans cette ville.

Traitement 3 : La maintenance des séjours est faite par les employés comme suit. Un employé peut détruire tous les séjours de date strictement inférieure à une date donnée. Paramètre(s) : jour. Retour(s) : le nombre de séjours détruits.

Traitement 4 : L'archivage d'un séjour est effectué automatiquement (pas par les employés ni les clients) en cas de destruction, avec l'avoir du client à ce moment. Paramètre(s) : néant. Retour(s) : néant.