

Université de Carthage Institut Préparatoire aux Etudes d'Ingénieur de Nabeul Département : Mathématiques		Année universitaire : 2022/2023
		Filière : SM/SP/ST
		Niveau d'étude : 2ème année
		Semestre : 2
		Nombre de pages : 2 pages
		Date : 02/03/2023, Durée : 1h30

DS Informatique N° 2

Un club de voyages propose des voyages organisés à ses adhérents. En septembre le club envoie à tous ses adhérents un catalogue contenant les voyages proposés pour l'année suivante.

Un voyage se caractérise par une référence unique, par sa destination, par sa durée (exprimée en nombre de semaines), par son coût, par la liste de toutes les dates de départ valides et par son type. Le type peut prendre les valeurs suivantes : « CT » pour circuit touristique, « RD » pour Raid-Découverte (voyage d'exploration et de découverte) et « SD » pour sport et détente.

Les adhérents peuvent s'inscrire à un ou plusieurs voyages ou adhérer uniquement pour profiter d'autres activités proposées par le club.

Toute personne qui participe à un voyage doit être adhérente au club.

Un numéro unique (code) est attribué à chaque adhérent (on conserve toujours les références des anciens adhérents de façon à ne pas refaire un dossier s'ils reviennent plusieurs années après).

La procédure d'attribution des codes est appelée « codification ». Pour attribuer des codes uniques aux différents adhérents on suit cette règle de codification :

Année d'adhésion au club + n° séquentiel

Exemple : le code 2023154 représente la personne numéro 154 adhérente à l'année 2023.

A chaque nouvelle année le code séquentiel recommence à partir de 1.

Les informations enregistrées dans chaque dossier sont le nom de l'adhérent, son prénom, son adresse et un indicateur (prend la valeur 1 ou 0) précisant s'il a réglé la cotisation de l'année en cours.

A chaque nouvelle année cet indicateur est mis à 0 pour tous les adhérents.

Une inscription à un voyage se fait pour une date de départ précise choisie parmi les dates valides et s'accompagne toujours du versement d'un acompte.

L'adhérent doit payer la totalité du coût du voyage (en un ou plusieurs paiements) deux jours avant la date de départ réservée, sinon son inscription à ce voyage sera annulée.

Pour gérer les adhésions au club et les inscriptions à des voyages une base de données est proposée selon le schéma relationnel suivant :

ADHERENT (NUMA, NOMA, PRENOMA, VILLEA, COT)
VOYAGE (REFV, DESTV, DUREEV, COUTV, TYPEV)
DEPART (#REFV, DATEDEP)
RESERVATION (NDOSSIER, #NUMA, #REFV, #DATEDEP, ACOMPTE)

Les clés primaires sont soulignées, les clés étrangères sont précédées par #.

NUMA : numéro d'adhérent, entier (ex : 202012)

NOMA : nom d'adhérent, chaîne de caractères

PRENOMA : prénom d'adhérent, chaîne de caractères

VILLEA : adresse (ville) d'adhérent, chaîne de caractères

COT : indicateur de règlement de la cotisation (1 si la cotisation de l'année en cours est réglée et 0 sinon)

REFV : référence du voyage, chaîne de caractères (ex : ALG12)

DESTV : destination du voyage, chaîne de caractères (ex : Algérie)

DUREEV : durée du voyage (ex : 2 pour deux semaines)

COUTV : coût du voyage (en DT)

TYPEV : type du voyage

DATEDEP : date de départ. La date devra toujours être enregistrée au même format « AAAA-MM-JJ » par exemple "2023-03-02".

NDOSSIER : numéro de dossier de réservation, entier

ACOMPTE : montant de l'acompte versé (en DT)

Travail demandé :

1. Décrire le rôle de la commande suivante écrite en Algèbre relationnelle.

$$R = \pi_{REFV, DESTV, COUTV} (\sigma_{DESTV='New York' \text{ et } DUREEV \geq 3} (VOYAGE))$$

2. Convertir, en Algèbre relationnelle, la commande SQL suivante permettant d'afficher les noms et prénoms des adhérents ayant réservé un voyage pour la destination "Japon" avec un acompte de plus de 2000 DT.

```
SELECT NOMA, PRENOMA
FROM ADHERENT
WHERE NUMA IN (SELECT DISTINCT NUMA
                FROM RESERVATION
                WHERE REFV IN (SELECT REFV
                              FROM VOYAGE
                              WHERE DESTV = 'Japon')
                AND ACOMPTE > 2000)
```

Répondre en langage **SQL** aux questions suivantes :

3. Combien d'adhérents ont réglé leur cotisation cette année ?
4. Combien d'adhérents se sont inscrits à un voyage pour chaque type de voyage ?
5. Que fait la requête SQL suivante et donner une autre version qui donne le même résultat.

```
SELECT      *
FROM        ADHERENT
WHERE       NUMA NOT IN (SELECT DISTINCT NUMA FROM RESERVATION)
```

6. Afficher les types de voyage ayant plus de 5 voyages organisés.
7. Mettre à jour le coût de tous les voyages qui durent plus de 3 semaines en ajoutant 10% au coût initial.
8. Retrouver le nombre de réservations, la somme et la moyenne des acomptes versés par voyage dont le nombre de réservation est supérieur ou égal à 2.
9. Quel est le voyage qui a reçu le plus grand nombre de réservations pour la date de départ '2023-07-01' ?
10. Quels sont les adhérents qui ont réservé tous les voyages proposés pour l'année prochaine ?
11. Quelle est la destination la plus populaire pour les adhérents qui ont réservé un voyage ?
12. Sélectionner les noms et prénoms des adhérents qui ont réservé un voyage pour la même date de départ qu'un autre adhérent.
13. Quel est le montant total des acomptes versés par les adhérents pour chaque voyage ?
14. Quels sont les voyages pour lesquels le montant total des acomptes versés est inférieur à la moitié du coût total du voyage ?
15. Supprimer toutes les réservations des adhérents qui n'ont pas réglé leur cotisation de l'année en cours.
16. Ajouter une colonne nommée *NB_PLACES_DISPONIBLE* à la table VOYAGE pour suivre le nombre de places disponibles et réservées par voyage. Le nombre de places est initialisé à 55 et décrémente de 1 à chaque réservation.

Indication

La requête SQL permettant d'ajouter une colonne à une table existante est la suivante :

ALTER TABLE nom_de_la_table ADD COLUMN nom_de_la_colonne type_de_données [Contraintes]

Exemple : ALTER TABLE etudiant ADD COLUMN email text

Cette commande SQL permet d'ajouter à la table etudiant une colonne nommée email de type text.

Après l'exécution de cette commande la table etudiant aura une nouvelle colonne 'email' qui sera initialisée à 'NULL' pour toutes les lignes existantes.

17. Ajouter une nouvelle réservation pour l'adhérent numéro 202312 pour un voyage de référence JAP456, avec une date de départ le 1er mai 2023 et un acompte de 1000 DT.
18. Supprimer les réservations pour les voyages qui ont déjà eu lieu.

Afin de faciliter la gestion des inscriptions aux voyages, suivre les paiements effectués par chaque adhérent, il pourrait être utile d'ajouter une table pour stocker les paiements des adhérents. Cette table pourrait avoir le schéma relationnel suivant :

PAIEMENT (ID, #RES, DATE_PAIEMENT, MONTANT_PAIEMENT)

Avec

- **ID** est la clé primaire de la table paiement, de type entier et s'incrémente automatiquement,
- **RES** est une clé étrangère faisant référence à la colonne *NDOSSIER* de la table *RESERVATION*,
- **DATE_PAIEMENT** est la date à laquelle le paiement a été effectué,
- **MONTANT_PAIEMENT** est le montant du paiement (en Dinars).

Les informations de cette table pourraient également être utilisées pour envoyer des rappels aux adhérents qui ont des paiements en retard et pour suivre les paiements en attente pour les voyages à venir.

19. Donner la requête SQL permettant de créer la table **PAIEMENT**.
20. Créer un script Python utilisant SQLite permettant d'extraire des informations sur les adhérents ayant des paiements en retard pour un voyage donné, et les écrire dans un fichier texte.

Le script demande la référence de voyage **ref_voyage** et utilise une requête SQL pour extraire les informations pertinentes de la base de données '**club_voyages.db**'.

Le fichier à créer aura comme nom : « retards_abc.txt » où « abc » est la référence de voyage ref_voyage.

Exemple : pour une ref_voyage égale à « ALG123 » le fichier sera « retards_ALG123.txt »

Chaque ligne du contenu du fichier doit suivre le format suivant : **Nom, Prénom, Reste à payer**

Indications

Quelques fonctions SQL utiles :

- **Date ()** : renvoie la date actuelle sous forme de chaîne de caractères dans le format "YYYY-MM-DD"
- **YEAR (date)** : permet d'extraire l'année d'une date donnée.
Exemple : YEAR ('2023-03-02') retourne 2023
- **DATEDIFF (day, date1, date2)** : calcul la différence entre deux dates en termes de nombre de jours (day). Exemple : DATEDIFF (day, '2022-08-25', '2021-08-25') retourne 365.
DATEDIFF peut être utilisée pour déterminer la différence entre deux dates en termes de nombre de mois (month), heures (hours), minutes, seconds, ...
- **DATE (date, '-2 days')** : soustrait 2 jours de la date spécifiée et renvoie le résultat en format Date.
Exemple : DATE ('2022-08-25', '-2 days') retourne '2022-08-23'.