

RÉVISIONS PL/SQL

Un club de spéléologie propose des sorties à ses adhérents. Pour effectuer ces sorties des paquetages de matériel (corde, casque, lampe ...) leur sont prêtés moyennant une caution.

Tables relationnelles:

```
create table ADHERENT (
  NO_ADHERENT      NUMBER(4) ,
  NOM_ADHERENT     VARCHAR2(30),
  PRENOM_ADHERENT  VARCHAR2(20),
  CREDITS           NUMBER(11,2) constraint adherent_credits_ck check
(credits > 0),
  constraint ADHERENT_pk primary key (NO_ADHERENT)
);
```

Table ADHERENT: l'attribut CREDITS représente une somme alimentée par l'adhérent. De CREDITS sont retirés le montant des participations aux sorties et la caution lors du prêt du paquetage. Important CREDITS doit toujours être positif.

```
create table PAQUETAGE (
  NO_PAQUETAGE     NUMBER(4),
  LIB_PAQUETAGE    VARCHAR2(30),
  CAUTION          NUMBER(11,2),
  constraint PAQUETAGE_pk primary key (NO_PAQUETAGE)
);
```

Table PAQUETAGE: CAUTION représente le montant de la caution à verser lors du prêt du paquetage.

```
create table PARTICIPE (
  NO_ADHERENT      NUMBER(4) ,
  NO_SORTIE        NUMBER(4) ,
  PAYEE           CHAR(1)      not null,
  constraint PARTICIPE_pk primary key (NO_ADHERENT, NO_SORTIE)
);
```

Table PARTICIPE: indique qu'un adhérent participe à une sortie, PAYEE contient 'F' si la participation n'a pas été payée et 'V' dans le cas contraire

```
create table PRET (
  DATE_PRET        DATE ,
  NO_PAQUETAGE     NUMBER(4) ,
  NO_ADHERENT      NUMBER(4)      not null,
  DATE_RETOUR      DATE,
  constraint PRET_pk primary key (DATE_PRET, NO_PAQUETAGE)
);
```

Table PRET: indique qu'à une date donnée un paquetage donné est prêté à un adhérent. La date de retour est à null tant que le paquetage n'a pas été rendu.

```
create table SORTIE (
  NO_SORTIE        NUMBER(4) ,
  LIBELLE_SORTIE   VARCHAR2(30),
  DATE_SORTIE      DATE,
  MT_PARTICIPATION NUMBER(11,2),
  constraint SORTIE_pk primary key (NO_SORTIE)
);
```

Table SORTIE: MT_PARTICIPATION, montant qui devra être payé par chacun des adhérents qui participera à cette sortie.

```
create table UTILISE (
  NO_PAQUETAGE      NUMBER(4) ,
  NO_SORTIE         NUMBER(4) ,
  constraint UTILISE_pk primary key (NO_PAQUETAGE, NO_SORTIE)
);
```

Table UTILISE: indique tous les paquetages prévus pour une sortie (un paquetage par adhérent potentiel).

Contraintes d'intégrité référentielles

```
alter table PARTICIPE
  add constraint PARTICIPE_NOSORTIE_fk foreign key (NO_SORTIE)
    references SORTIE
  add constraint PARTICIPE_NOADHERENT_fk foreign key (NO_ADHERENT)
    references ADHERENT;

alter table PRET
  add constraint PRET_NOADHERENT_fk foreign key (NO_ADHERENT)
    references ADHERENT
  add constraint PRET_NOPAQUETAGE_fk foreign key (NO_PAQUETAGE)
    references PAQUETAGE;

alter table UTILISE
  add constraint UTILISE_NOPAQUETAGE_fk foreign key (NO_PAQUETAGE)
    references PAQUETAGE
  add constraint UTILISE_NOSORTIE_fk foreign key (NO_SORTIE)
    references SORTIE;
```

Écrire le code PL/SQL pour répondre aux questions suivantes:

1. L'adhérent peut à tout moment verser une somme pour alimenter son crédit. Ecrire la procédure permettant d'enregistrer ce versement. Gérer les exceptions.
2. Pour éviter au trésorier de manipuler trop d'argent on souhaite prélever automatiquement le montant de la participation (MT_PARTICIPATION) à une sortie du crédit de l'adhérent (CREDITS) dès que la colonne PAYEE de la table PARTICIPE passe de l'état 'F' à 'V'. Indiquer et éventuellement coder le ou les dispositifs logiciels permettant de répondre au problème posé. Gérer les exceptions.
3. Pour effectuer une sortie, le club prête à chaque adhérent un paquetage (corde, casque, lampe ...) moyennant le dépôt d'une caution qui est prélevée sur son crédit. Développer la fonction generer_prets(pnumsortie sortie.no_sortie%type) qui affecte un paquetage à chaque participant à une sortie et prélève sur son crédit le montant de la caution.

Remarques

- ✓ Si l'adhérent n'a pas assez de crédits pour payer la caution du paquetage cela ne devra pas interrompre le traitement général, mais devra être signalé par le message suivant : 'Problème de caution pour le paquetage N et l'adhérent M'. N et M étant les numéros. Le prêt ne pourra donc pas être réalisé pour cet adhérent.
(Si besoin, le numéro d'erreur Oracle associé à la non vérification d'une contrainte *check* est -2290).
- ✓ Toute autre erreur fatale interrompra immédiatement le traitement, invalidera les transactions, fera l'objet d'un message indiquant le type d'erreur, et la fonction retournera -1.
- ✓ Il est possible qu'il n'y ait pas assez de paquetages prévus et réellement disponibles pour tous les participants effectivement inscrits. Si cela survient, on interrompra le traitement, les transactions seront validées, un message indiquera qu'il n'y a pas assez de paquetages et la fonction retournera -2.
- ✓ Si tout s'est bien passé les transactions seront validées et la fonction retournera 0.
- ✓ Pour afficher les messages, on utilisera `dbms_output.put_line()`.
- ✓ L'algorithme général est le suivant :
Pour chaque participant ayant payé sa participation à la sortie passée en paramètre
Récupérer le premier paquetage disponible (date de retour non nulle et prévu pour la sortie),
Affecter ce paquetage à l'adhérent en cours à la date du jour si le crédit de l'adhérent est suffisant (décrémenter le montant de la caution du crédit de l'adhérent).

Travail à réaliser:

- le code source
- le (les) test(s) prouvant le bon fonctionnement du code