

Exercice 1:

On considère le schéma de base suivant où les clés primaires sont soulignées, et les clés étrangères sont précédées d'un #.

Ville (noville , nomville , codepostal)

Train (notrain , date_dep , date_arr , #noville_dep , #noville_arr)

Passager (nopassager , Nom , #notrain , #date_dep)

Tarif (#notrain , #date_dep , categorie , prix)

Distance (#noville1 , #noville2 , nbkm)

Commentaires : *noville_dep*, *noville_arr*, *noville1* et *noville2* se réfèrent à *noville* de la table Ville.

Questions :

(a) Donner les commandes SQL d'Oracle pour créer les tables ci-dessus, en spécifiant les clés primaires et étrangères, et en assurant que :

- Les prix de la table Tarif et les distances entre les villes soient des nombres positifs ;
- Pour un numéro de train, le numéro de la ville d'arrivée soit différent de celui la ville de départ.

```
CREATE TABLE Ville (
  noville NUMBER PRIMARY KEY,
  nomville VARCHAR2(30),
  codepostal NUMBER
);

CREATE TABLE Train (
  notrain NUMBER,
  date_dep DATE,
  date_arr DATE,
  noville_dep NUMBER,
  noville_arr NUMBER,

  CONSTRAINT cTr_1 CHECK (noville_arr<>noville_dep),
  CONSTRAINT cTr_2 PRIMARY KEY (notrain,date_dep),
  CONSTRAINT cTr_3 FOREIGN KEY (noville_dep)
    REFERENCES Ville(noville),
  CONSTRAINT cTr_4 FOREIGN KEY (noville_arr)
    REFERENCES Ville(noville)
);

CREATE TABLE Passager (
  nopassager NUMBER PRIMARY KEY,
  nom VARCHAR2(30),
  notrain NUMBER,

  CONSTRAINT cPass_1 FOREIGN KEY (notrain,date_dep)
    REFERENCES Train(notrain,date_dep)
);

CREATE TABLE Tarif (
  notrain NUMBER,
  categorie NUMBER,
  prix NUMBER CONSTRAINT cTar_1 CHECK (prix>=0),

  CONSTRAINT cTar_2 PRIMARY KEY (notrain,categorie),
  CONSTRAINT cTar_3 FOREIGN KEY (notrain,date_dep)
    REFERENCES Train(notrain,date_dep)
);
```

```

CREATE TABLE Distance (
  noville1 NUMBER,
  noville2 NUMBER,
  nbkm NUMBER CONSTRAINT cDist_1 CHECK (nbkm>=0),

  CONSTRAINT cDist_2 PRIMARY KEY (noville1,noville2),
  CONSTRAINT cDist_3 FOREIGN KEY (noville1)
    REFERENCES Ville(noville),
  CONSTRAINT cDist_4 FOREIGN KEY (noville2)
    REFERENCES Ville(noville)
);

```

- (b) Définir un trigger pour assurer qu'à l'insertion de données dans la table Train la différence entre la date d'arrivée et la date de départ d'un train ne dépasse pas 8 heures si la distance entre la ville de départ et la ville d'arrivée ne dépasse pas 300Km.

```

CREATE OR REPLACE TRIGGER trigger_temps
  BEFORE INSERT ON Train
  FOR EACH ROW
  DECLARE Dist NUMBER;
  BEGIN
    SELECT DISTINCT nbkm INTO Dist
    FROM Distance
    WHERE noville1=:NEW.noville_dep AND noville2=:NEW.noville_arr;

    IF (Dist<300 AND NEW.date_arr-NEW.date_dep>=8/24)
    THEN
      :NEW.date_arr := :NEW.date_dep + 8/24;
    END IF;
  END;

```

On peut aussi écrire le trigger en utilisant les exceptions :

```

CREATE OR REPLACE TRIGGER trigger_temps
  BEFORE INSERT ON Train
  FOR EACH ROW
  DECLARE
    Dist NUMBER;
    Erreur EXCEPTION;
  BEGIN
    SELECT DISTINCT nbkm INTO Dist
    FROM Distance
    WHERE noville1=:NEW.noville_dep AND noville2=:NEW.noville_arr;

    IF (Dist<300 AND NEW.date_arr-NEW.date_dep>=8/24)
    THEN
      RAISE Erreur;
    END IF;

  EXCEPTION
    WHEN Erreur THEN
      RAISE_APPLICATION_ERROR(-20322,'La duree du trajet est
      superieure a 8h pour une distance entre les 2 villes
      inferieur a 300km');

  END;

```

Exercice 2:

Considérer les schémas de relations suivants :

EMP (N , S , E , P , Dp) // N est un employé dans l'équipe E, il a un salaire mensuel S, et participe a un projet P à partir de la date Dp.

CHEF (C , E , P , Dc) // L'employé C est le chef de l'équipe E, qui crée un projet P a la date Dc.

Sur les schémas EMP et CHEF, on considère les contraintes suivantes:

- (i) Chaque équipe a un seul chef, et chaque chef est le chef d'une seule équipe. Chaque projet a une seule date de création.

Chaque équipe a un seul chef : $E \rightarrow C$

Chaque chef est le chef d'une seule équipe : $C \rightarrow E$

Chaque projet a une seule date de création : $P \rightarrow Dc$

- (ii) Chaque employé a un seul salaire mensuel, et est dans une seule équipe. La date de début de participation d'un employé à un projet est unique.

Chaque employé a un seul salaire mensuel et est dans une seule équipe : $N \rightarrow S,E$

La date de début de participation d'un employé à un projet est unique : $P,E \rightarrow Dp$

- (iii) Une équipe qui figure dans la table EMP doit exister dans la table CHEF. Aussi un projet qui figure dans la table EMP doit exister dans la table CHEF.

Questions :

- (a) Donner les clés des schémas EMP et CHEF.

Schéma EMP :

- Clé primaire $\rightarrow N,P$
- Clé étrangère $\rightarrow (E,P)$
- Clé unique $\rightarrow N$

Schéma CHEF :

- Clé primaire $\rightarrow (E,P)$
- Clés uniques $\rightarrow E, C$ et P

- (b) En utilisant SQL d'Oracle et les contraintes clés primaires, uniques, et étrangères donner les commandes pour créer les tables EMP et CHEF.

```
CREATE TABLE CHEF (  
  C VARCHAR2(30) UNIQUE NOT NULL,  
  E VARCHAR2(30) UNIQUE NOT NULL,  
  P VARCHAR2(30) UNIQUE NOT NULL,  
  Dc DATE,  
  
  CONSTRAINT cChef1 PRIMARY KEY (E,P)  
);  
  
CREATE TABLE EMP (  
  N VARCHAR2(30) UNIQUE NOT NULL,  
  S NUMBER,  
  E VARCHAR2(30),  
  P VARCHAR2(30),  
  Dp DATE  
  
  CONSTRAINT cEmp1 PRIMARY KEY (N,P)  
  CONSTRAINT cEmp2 FOREIGN KEY (E,P) REFERENCES CHEF (E,P)  
);
```

- (c) En plus des contraintes ci-dessus, on doit assurer que la date de participation d'un employé à un projet doit être ultérieure ou égale à la date de la création du projet. Définir un trigger pour renforcer cette contrainte.

```
CREATE OR REPLACE TRIGGER Date_Participation
BEFORE INSERT ON EMP
FOR EACH ROW
BEGIN
    SELECT DISTINCT Dc INTO :NEW.Dp
    FROM CHEF
    WHERE E=:NEW.E AND P=:NEW.P AND Dc>:NEW.Dp;
END;
```

On peut aussi écrire le trigger en utilisant les exceptions :

```
CREATE OR REPLACE TRIGGER Date_Participation
BEFORE INSERT ON EMP
FOR EACH ROW
DECLARE
    Date_Creation NUMBER;
    Erreur EXCEPTION;
BEGIN
    SELECT DISTINCT Dc INTO Date_Creation
    FROM CHEF
    WHERE E=:NEW.E AND P=:NEW.P AND Dc>:NEW.Dp;

    IF (Date_Creation>:NEW.Dp)
    THEN
        RAISE Erreur;
    END IF;

EXCEPTION
    WHEN Erreur THEN
        RAISE_APPLICATION_ERROR(-20322,'La date de participation ne
        peut pas etre anterieure a la date de creation');

END;
```