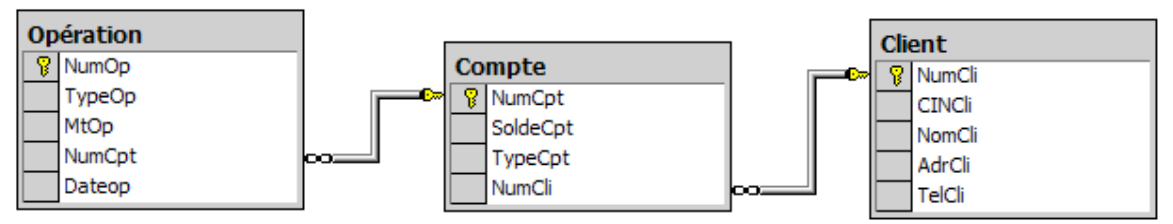
**Exercice T2** Soit la base de données GestionCompte suivante :



Les opérations consistent en des opérations de retrait ou de dépôt d'argent (TypeOp=D si le client a déposé de l'argent sur son compte et TypeOp=R si le client a retiré de l'argent sur son compte).

Un client ne peut avoir qu'un seul compte courant (TypeCpt="CC") et qu'un seul compte sur carnet (TypeCpt="CN")

Le numéro d'opération est automatique.

La date de l'opération prend par défaut la date du jour.

**Créer les triggers suivants**

TR 1. Qui à l'ajout ou à la modification de clients dans la table client vérifie si les Numéros de CIN saisies n'existent pas pour d'autres clients.

|  |
| --- |
| Méthode 1 :  /\*Qui à l'ajout de clients dans la table client vérifie si les Numéros de CIN saisies n'existent pas pour d'autres clients\*/  delimiter $$  CREATE trigger tr1\_Ajout\_CIN BEFORE insert on client for each ROW  begin  declare x varchar(8);  set x= new.CinCli;  if EXISTS (select \* from client where cinCli=x) THEN  signal SQLSTATE '45000'  set MESSAGE\_TEXT " Ce CIN existe déjà pour un autre client";  end if;  end$$ |
| Méthode 2 :  delimiter $$  CREATE trigger tr1\_Ajout\_CIN2 BEFORE insert on client for each ROW  begin  declare x varchar(8);/\*X : une variable pour récupérer le CIN du client à ajouter\*/  declare nbr int;/\*nbr : une variable pour récupérer du nombre des CIN dans la BD\*/  set x= new.CinCli;/\* recupération du CIN à partir du new\*/  select count(CinCli) into nbr from client where CinCli=x;  if nbr>0 THEN  signal SQLSTATE '45000'  set MESSAGE\_TEXT=" Ce CIN existe déjà pour un autre client: méthode2";  end if;  end$$ |

TR 2. Qui à la création de comptes, vérifie si :

* Les soldes sont supérieurs à 1500 DH ;
* Les types de compte sont CC ou CN et aucune autre valeur n'est acceptée ;
* Les clients n'ont pas déjà des comptes du même type.

|  |
| --- |
| DELIMITER $$  create trigger tr2\_Create\_Cpt BEFORE insert on compte for EACH row  BEGIN  declare NCpt int; /\*Var pour récupérer le numéro du compte à créer\*/  declare Sld float; /\*Var pour récupérer le solde du compte à créer\*/  declare NCli int; /\*Var pour récupérer le numéro du client pour lequel on va créer le compte\*/  declare TypCpt varchar(2); /\*Var pour récupérer le type du compte à créer\*/  declare nbr int;/\*var pour stocker le nombre de compte de meme type possédé par le cleint\*/  set NCpt= new.NumCpt;  set NCli=new.NumCli;  set TypCpt= new.typeCpt;  set sld= new.soldeCpt;  /\*-------------Les soldes sont supérieurs à 1500 DH ;-------------------\*/  if sld<1500 THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Le solde doit etre supérieur à 15000";  end IF;  /\*-------------Les types de compte sont CC ou CN et aucune autre valeur n'est acceptée ;\*/  if TypCpt not in ('CC','CN') THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Le type du compte ne peut etre que CC ou CN"  end if  /\*-----------Le client n'a pas déjà des comptes du même type.-----------------\*/  select count(NumCpt) into nbr from compte where numCli=Ncli and typeCpt = TypCpt;  if nbr >0 THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Ce client a déjà un autre compte de meme type";  end if;  end$$ |
| CREATE TRIGGER `tr2\_Create\_Cpt` BEFORE INSERT ON `compte`  FOR EACH ROW BEGIN  /\*-------------Les soldes sont supérieurs à 1500 DH ;-------------------\*/  if new.soldeCpt<1500 THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Le solde doit etre supérieur à 15000";  end IF;  /\*-------------Les types de compte sont CC ou CN et aucune autre valeur n'est acceptée ;\*/  if new.typeCpt not in ('CC','CN') THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Le type du compte ne peut etre que CC ou CN";  end if;  /\*-----------Le client n'a pas déjà des comptes du même type.-----------------\*/  if exists (select \* from compte where numCli=new.numCli and typeCpt= new.typeCpt ) THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Ce client a déjà un autre compte de meme type";  end if;  end |
| CREATE TRIGGER `tr2\_Create\_Cpt` BEFORE INSERT ON `compte`  FOR EACH ROW BEGIN  declare valide boolean DEFAULT true;  /\*-------------Les soldes sont supérieurs à 1500 DH ;-------------------\*/  if new.soldeCpt<1500 THEN  valide =false;  end IF;  /\*-------------Les types de compte sont CC ou CN et aucune autre valeur n'est acceptée ;\*/  if new.typeCpt not in ('CC','CN') THEN  valide= false;  end if;  /\*-----------Le client n'a pas déjà des comptes du même type.-----------------\*/  if exists (select \* from compte where numCli=new.numCli and typeCpt= new.typeCpt ) THEN  valide=false;  end if;  if valide=false THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT ="Les données saisies ne sont pas correctres, vérifier le solde>1500,Type 'CC' ou 'CN', si le client n'as pas déjà un autre compte de meme type";  end if;  end |

TR 3. Qui à l'ajout ou à la modification d’une opération :

* le typeOp ne peut être que ‘’R’’ ou ‘‘D’’
* Le numéro du compte doit exister dans la table compte.
* La date de l’opération doit être inférieure ou égale à la date d’aujourd’hui

|  |
| --- |
| DELIMITER $$  create trigger tr3\_Create\_Op BEFORE insert on operation for EACH row  BEGIN  if new.TypeOp not in ('R','D') THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Le type du operation ne peut etre que R ou D";  end if;  if not EXISTS (SELECT NumCpt FROM compte c WHERE NumCpt = new.NumCpt ) THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Le numéro du compte doit exister dans la table compte";  end if;    if new.DateOp>CURRENT\_DATE() THEN  set MESSAGE\_TEXT="La date doit etre inférieure ou égale à la date d’aujourd’hui";  end if;  end$$ |

TR 4. Qui interdit la suppression de comptes dont le solde est > 0 ou de comptes pour lesquels la dernière opération date de moins de 3 mois même s'ils sont vides (solde=0).

|  |
| --- |
| create trigger tr4\_Delete\_Op BEFORE delete on compte for EACH row  BEGIN  declare nbrj int;  if old.soldeCpt > 0 THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="On ne peut pas supprimer ce compte, car son solde est >0";  end if;  if old.soldeCpt = 0 THEN  select Min(datediff(curDate(),DateOp)) into nbrj FROM operation where numCpt=old.NumCpt;  if nbrj<90 then  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="On ne peut pas supprimer ce compte, il a une opération de moins de 3 mois";  end if;  end if;  end; |

TR 5. Qui :

Interdit d’ajouter une opération retrait de type ‘’R’’ sur un comptes si le montant de l’opération est supérieur su solde du compte ;

|  |
| --- |
| BEGIN  DECLARE NCpt int;  declare Montant, solde float;  DECLARE typOp varchar(1);  set NCpt= new.NumCpt;/\* le numéro du compte sur lequel on va faire l'opération\*/  set Montant = new.MntOp;/\* le montant de l'opération à éffectuer\*/  SELECT soldeCpt into solde from compte where NumCpt=Ncpt; /\* Récupérer le sode du compte sur lequel on va faire l'opération \*/  set typOp =new.TypeOp;/\*récupérer le type de l'opération\*/  if typOp ='R' THEN  begin  if montant > solde THEN  SIGNAL SQLSTATE '45000'  set MESSAGE\_TEXT="Le solde est insuffisant pour ce compte";  end if;  end;  end if;  end |

TR 6. Qui :

Interdit la modification des numéros de comptes ;

Interdit la modification du solde de comptes auxquels sont associées des opérations ;

Ne permet pas de faire passer des comptes sur carnet en comptes courants, le contraire étant possible;

TR 7. A la modification de numéros de clients dans la table compte, vérifie si les nouveaux clients n'ont pas de comptes associés du même type.

TR 8. A la création d’une nouvelle opération :

* Incrémenter le solde du compte concerné par le montant de l’opération si le type est ‘’D’’
* Décrémenter le solde du compte concerné par le montant de l’opération si le type est ‘’R’’