

```
/*Créer un trigger qui assure q'un nouveau client a un CIN unique*/
```

```
delimiter $$
```

```
CREATE TRIGGER Tr1 before insert on client
```

```
for each ROW
```

```
BEGIN
```

```
set @cin=new.CINCl;
```

```
set @nbr=(select count(*) from client where CINCl=@cin);
```

```
if @nbr=1 THEN
```

```
    SIGNAL SQLSTATE '40000'
```

```
    set MESSAGE_TEXT ='Ce numero CIN existe déjà';
```

```
end if;
```

```
end$$
```

```
CREATE TRIGGER `Tr1` AFTER INSERT ON `client` FOR EACH ROW
```

```
BEGIN
```

```
set @cin=new.CINCl;
```

```
set @nbr=(select count(*) from client where CINCl=@cin);
```

```
if @nbr=2 THEN
```

```
    SIGNAL SQLSTATE '40000'
```

```
    set MESSAGE_TEXT ='Ce numero CIN existe déjà';
```

```
end if;
```

```
end
```

```
DROP TRIGGER IF EXISTS `Tr1`;
```

```
CREATE TRIGGER `Tr1` before INSERT ON `client`
```

```
FOR EACH ROW BEGIN
```

```
set @cin=new.CINCl;
```

```
/*set @nbr=(select count(*) from client where CINCl=@cin);
```

```
if @nbr=2 THEN*/
```

```
/*if @cin in (select CinCl from client) then .... end if;*/
```

```
if EXISTS (select * FROM client where CINCl=@cin ) then
```

```
    SIGNAL SQLSTATE '40000'
```

```
    set MESSAGE_TEXT ='Ce numero CIN existe déjà';
```

```
end if;
```

```
end
```

```
delimiter $$
```

```
CREATE TRIGGER Tr10 before update on client
```

```
for each ROW
```

```
BEGIN
```

```
set @cin=new.CINCl;
```

```
/*verifier que le CIn a été modifié*/
```

```
if new.CINCl != old.CINCl then
```

```
if @cin in (select CINCl from client) THEN
```

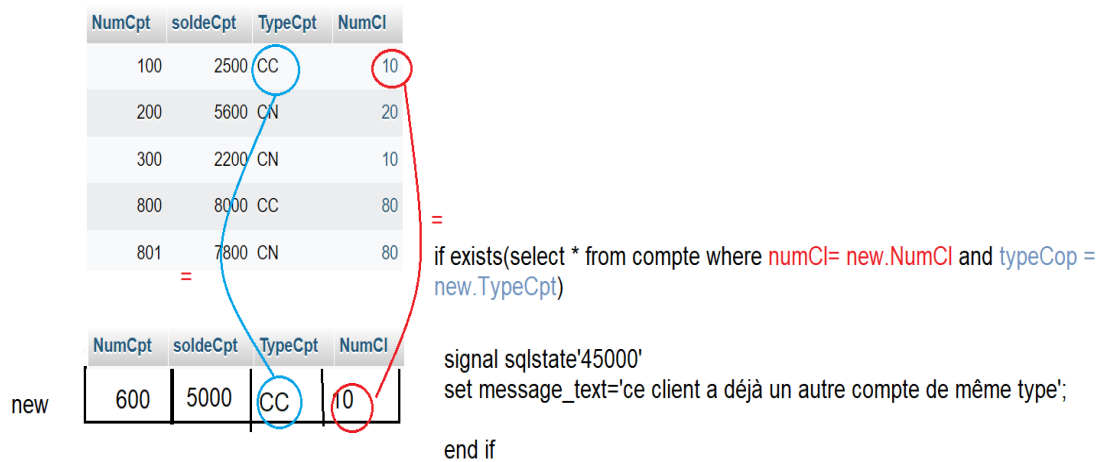
```
    SIGNAL SQLSTATE '40000'
```

```
    set MESSAGE_TEXT ='Erreur de modification !!! Ce numero CIN existe déjà';
```

```
end if;
```

```
end if;
```

```
endend$$
```



```

/*Lors de la création du compte*/
delimiter $$
create TRIGGER tr2 before insert on compte
for EACH ROW
BEGIN
DECLARE s float;
declare typec varchar(2);
/*Vérifier que le solde >1500*/
set s= new.soldeCpt; /*récupérer le solde du compte à créer*/
set typec=new.typeCpt; /*récupérer le type du compte à créer*/
IF s<1500 THEN
    SIGNAL SQLSTATE '30000'
    set MESSAGE_TEXT='Le solde doit être >1500';
end if;
/*Vérifier que le type compte est CC ou CN */
if typec !='CC' and typeC != 'CN' THEN
    signal SQLSTATE '30000'
    set MESSAGE_TEXT='Le type du compte doit être CC ou CN';
end if;
/* Vérifier que le client ne possède pas un autre compte de même type*/

if exists (select * from compte where numCl=new.numcl and TypeCpt= new.typeCpt) then
    SIGNAL SQLSTATE '30000'
    set MESSAGE_TEXT ='Ce client possède déjà un autre compte de même type';
end if;

end$$

```

```

DELIMITER $$
drop trigger if exists tr2;$$
create TRIGGER if not exists tr2 before insert on compte
for EACH ROW
BEGIN
DECLARE s float;
declare typec varchar(2);
/*-----*/
set @valide=1;
set @erreur="";
/*-----*/

```

```

/*Vérifier que le solde >1500*/
set s= new.soldeCpt; /*récupérer le solde du compte à créer*/
set typec=new.typeCpt; /*récupérer le type du compte à créer*/
IF s<1500 THEN
    set @valide=-1;
    set @erreur='Le solde doit être >1500';
end if;
/*Vérifier que le type compte est CC ou CN */
if typec !='CC' and typeC != 'CN' THEN
    set @valide=-1;
    set @erreur=concat(@erreur,'\n','Le type du compte doit être CC ou CN');
end if;
/* Vérifier que le client ne possède pas un autre compte de même type*/

if exists (select * from compte where numCl=new.numcl and TypeCpt= new.typeCpt) then
    set @valide=-1;
    set @erreur =concat(@erreur,'\n','Ce client possède déjà un autre compte de même type');
end if;

if @valide=-1 THEN
    signal SQLSTATE '40000'
    set MESSAGE_TEXT=@erreur;
end if;

end$$

delimiter $$
drop trigger if exists tr3;$$
create TRIGGER if not exists TR3 BEFORE delete on compte
for EACH ROW
BEGIN

set @s= old.soldeCpt;
if @s>0 THEN
    SIGNAL SQLSTATE '40000'
    set MESSAGE_TEXT='On ne peut pas supprimer un compte si le solde est>0';
end if;

if exists(select * from operation where TimeStampDiff(month, dateOp , now())<3 and Numcpt=
old.numCpt) THEN
    signal SQLSTATE '40000'
    set MESSAGE_TEXT='On ne peut pas supprimer ce copte car il possède des opérations récentes';
end if;
delete from operation where NumCpt=old.numCpt;
end;

delimiter $$
drop trigger if exists TR4;$$
create TRIGGER if not exists TR4 BEFORE update on compte
for EACH ROW
BEGIN

/*-----vérifier que le numero du compte n'a pas été modifié*/

```

```

if new.numCpt != old.NumCpt THEN
    SIGNAL SQLSTATE '40000'
    set MESSAGE_TEXT='On ne peut pas modifier le numéro du compte';
end if;
/*-----on ne peut pas modifier le solde si le compte a des opérations associées*/
if new.soldeCpt !=old.SoldeCpt then
if exists(select * from operation where Numcpt= new.numCpt ) THEN
    signal SQLSTATE '40000'
    set MESSAGE_TEXT='On ne peut pas modifier le solde du compte car il possède des opérations';
end if;
end if;
/*-----On ne peut pas passer le type du compte du CN --> CC*/
if old.typeCpt= 'CN' and new.Typecpt='CC' then
    signal sqlstate '40000'
    set MESSAGE_TEXT ='n ne peut pas passer le type du compte du CN --> CC';
end if;
/*-----Vérifier que le client n'a pas un autre compte de même type*/
if new.NumCl != old.NumCl then
if exists(select * from compte where numcl=new.NumCl and typeCpt= new.typeCpt) then
    SIGNAL SQLSTATE '40000'
    set MESSAGE_TEXT='Ce client a déjà un compte de même type';
end if;
end if;
end$$

```