Rappel : Lorsque Le SGBD Mysql exécute les opérations de Manipulation de données, il crée des pseudo-tables :

**New** : pseudo table pour stocker les données à ajouter dans la table en attendant l’exécution des triggers liés à l’événement INSERT

**Old :** pseudo table créé pour stocker les données à supprimer en attendant l’exécution des triggers liés à l’événement DELETE

|  |  |
| --- | --- |
| Action LMD | Pseudo tables créés |
| Insert | new |
| delete | old |
| Update (combinaison de :Delete des anciennes valeur + Insert de nouvelle valeurs) | New + old |

Triggers sur la BD ventes

1. Créer un trigger qui va empêcher de saisir un article avec le prix <0

DELIMITER $$

create trigger TR\_Prix **BEFORE** INSERT on **article** for EACH ROW

BEGIN

if new.PU <0 THEN

SIGNAL SQLSTATE '45000'

set MESSAGE\_TEXT=' Le prix ne peut pas être négatif ';

end if;

END

$$

1. Créer un trigger qui empêche d’ajouter un article si le SeuilMin > SeuilMax

create TRIGGER trStock BEFORE INSERT on article for each ROW

BEGIN

if new.Stock\_Min > new.Stock\_Max THEN

SIGNAL SQLSTATE '45000'

set MESSAGE\_TEXT='Le stock min ne doit pas etre supérieur au stock max';

end if;

END;

1. Créer un trigger qui empêche de modifier le stock d’un article.

create TRIGGER tr\_stock **BEFORE** UPDATE on **article** for each ROW

BEGIN

if old.Qte\_stock != new.Qte\_stock THEN

SIGNAL SQLSTATE '45000'

set MESSAGE\_TEXT='On ne peut pas modifier la quantité en stock';

end if;

end;

1. Créer un trigger qui va modifier la quantité en stock lors de l’ajout de lignes dans la table Ligne\_commande

DELIMITER $$

CREATE TRIGGER Tr\_Vente **AFTER** INSERT on **detail\_cmd** for each ROW

BEGIN

declare codeA , qtt , stock int;

/\*récupérr le code de l'article à vendre de la ligne detail\_cmd\*/

set codeA= new.code\_art;

/\*récupérr la quantité à vendre de la ligne detail\_cmd\*/

set qtt= new.Qté\_comdé;

select Qte\_stock into stock from article where code\_art=codeA;

/\*savior si on a suffisament de quantité en stock\*/

if stock <qtt THEN

SIGNAL SQLSTATE '45000'

set MESSAGE\_TEXT='La quantié en stock n est pas suffisante pour faire cette vente';

ELSE

begin

/\*mettre à jour le stock \*/

**update** **article** set Qte\_stock= Qte\_stock-qtt where code\_art= codeA;

end;

end if;

end$$

1. Créer un trigger qui empêche de modifier l’Email du client par une valeur qui existe déjà pour un autre client

create trigger tr\_email BEFORE UPDATE on client for EACH ROW

BEGIN

DECLARE em varchar(60);

/\* 1- Récupérer la nouvelle valeur de l'email \*/

/\* 2- verifier si cette valeur n'existe pas chez un autre client \*/

set em= new.Email;

if EXISTS (select \* from client where Email like em) THEN

BEGIN

SIGNAL SQLSTATE '45000'

set MESSAGE\_TEXT='Cet email existe déjà cher un autre client';

end;

end if;

end;

1. Créer un trigger qui empêche d’ajouter un client avec codev qui n’existe pas dans la table ville.

DELIMITER $$

create trigger te\_ville BEFORE insert on client for EACH ROW

BEGIN

DECLARE v int;

/\* 1- Récupérer la nouvelle valeur de code de la vile\*/

/\* 2- verifier si cette valeur existe dans la table ville \*/

set v= new.CodeV;

if not EXISTS (select \* from ville where CodeV like v) THEN

BEGIN

SIGNAL SQLSTATE '45000'

set MESSAGE\_TEXT="Cette valeur de code ville n'existe pas dans la table ville";

end;

end if;

end

$$

1. Créer un trigger qui vérifier si le code postal de la ville est sous le format XNENNN  
   X est un chiffre 1-9  
   N est un chiffre 0-9  
   E est un espace
2. Créer un trigger qui vérifier si le CIN d’un soit sous la forme XYNNNNNN

X est une lettre obligatoire

Y est une lettre ou un espace  
N est un chiffre 0 à 9