

---

## OBJECTIF

---

### Objectif :

- Mettre en œuvre des triggers

### Ressources :

- Vous disposez du diagramme UML et du modèle physique de la base Produit sur laquelle vous devez vous appuyer pour réaliser les fonctions.
- diaporama de cours.

---

## BASE UTILISEE : PRODUIT

---

1. Lancez la console d'administration du SGBD PostGres :  
`http://a-pedagoarles-postgresql.aix.univ-amu.fr/phppgadmin/`  
login AMU / mot de passe AMU
2. Récupérez le script de création des tables (*creation\_produit.sql*), copier/coller dans la fenêtre SQL
3. lancer le script
4. Vos tables sont "normalement" créées et renseignées dans le schéma de données TP\_Produit, vérifiez le en utilisant la partie gauche de phppgadmin

*Vous pouvez commencer le TP, à vous de jouer!!*

---

## TRAVAIL A FAIRE

---

### Objectif :

Vous travaillez sur une base de données commerciale contenant les tables suivantes: produit, commande et ligne de commande. Il s'agit de réaliser des triggers permettant la mise à jour automatiquement de tous les montants, ainsi que les quantités en stock.

On ne pourra pas enregistrer une ligne de commande que l'on ne pourrait pas satisfaire pour cause de stock insuffisant.

Il s'agit d'une vue partielle de l'application. Notez bien que l'on ne gère pas ici les clients, ni les entrées en stock.

---

### CREER LES TRIGGERS QUI PERMETTENT DE SATISFAIRE LES REGLES SUIVANTES

---

1. **Lorsqu'il y a nouvelle ligne de commande**, le numéro de ligne de commande (num\_ligne) doit être incrémenté, il dépend de la commande, la quantité en stock du produit concerné est diminuée. Le montant de la commande est augmenté du montant de la ligne, cad de la quantité par le prix unitaire du produit.

**ATTENTION on ne doit pas pouvoir ajouter une ligne de commande (ligne\_cde) à une commande (commande) alors qu'il existe le même produit sur une autre ligne, dans ce cas message d'erreur.**

Vous testerez le trigger avec les instructions SQL suivantes :

```
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(100,15,10); -- pas assez de produit
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(100,15,60); -- le produit n'existe pas
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(100,5,10); -- test du trigger SANS erreur
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(100,5,10);      -- le produit existe déjà
```

Vérifiez que :

- le Numéro de la ligne de commande est cohérent,
- le montant de la commande est modifié,
- la quantité en stock du produit est à jour.

## 2. trigger sur la modification

- 2.1. A chaque mise à jour d'une ligne de commande il faut mettre à jour le montant de la commande et les quantités en stock. On suppose que seule la quantité peut être modifiée sur une ligne de commande et plus ou en moins. Vous ferez aussi attention au stock du produit !

**Vous testerez le trigger avec les instructions SQL suivantes :**

-- Insertion d'une ligne de commande pour le produit N°20 et la commande N°100

```
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(100,5,20);
```

-- test du trigger d'update AVEC erreur -- **ERREUR: Pas assez de stock pour le produit 20**

```
UPDATE LIGNE_CDE
```

```
SET QTE=20
```

```
WHERE NO_CDE=100
```

```
AND NO_PDT=20;
```

-- test du trigger d'update SANS erreur (produit en plus par rapport à avant)

```
UPDATE ligne_cde
```

```
SET qte=10
```

```
WHERE no_cde=100
```

```
AND no_pdt=20;
```

Vérifier que la ligne de commande a été modifiée (qte=10)  
que le stock du produit est modifié (stock = 0)  
que le montant de la commande est modifié  
(augmentation du montant)

-- test du trigger d'update SANS erreur (produit en moins par rapport à avant)

```
UPDATE ligne_cde
```

```
SET qte=5
```

```
WHERE no_cde=100
```

```
AND no_pdt=20;
```

Vérifier que la ligne de commande a été modifiée (qte=5)  
que le stock du produit est modifié (stock = 5)  
que le montant de la commande est modifié (diminution  
du montant)

- 2.2. Reprenez le trigger du 1 (suppression) et le modifier pour que quand le produit existe déjà, plutôt que de d'envoyer message d'erreur, faire une mise à jour de la ligne de commande

Testez votre trigger

3. **En cas de suppression** d'une ligne de commande, le stock du produit doit être mise à jour ainsi que le montant de la commande.

Vous ferez attention à laisser les numéros de lignes de commande dans un état cohérent, cad sans trou (1,2,4,5....)

-- test du trigger

-- ajout d'une nouvelle commande et de ces lignes de commande

```
INSERT INTO commande VALUES(300,now(),0);
```

```
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(300,2,50);
```

```
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(300,2,10);
```

```
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(300,5,40);
```

```
INSERT INTO ligne_cde(no_cde,qte,no_pdt) VALUES(300,4,30);
```

-- suppression d'une ligne dans la table ligne\_cde

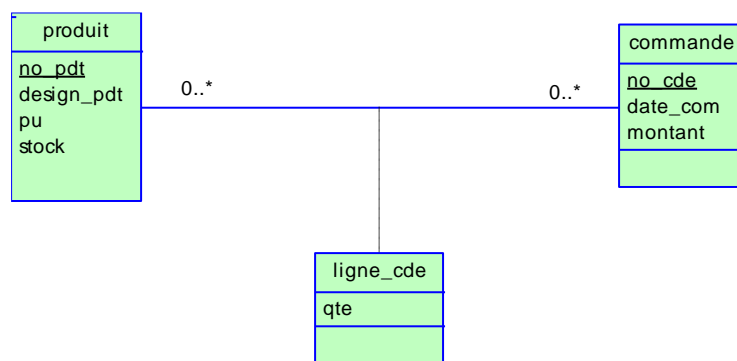
```
DELETE FROM ligne_cde
```

```
WHERE no_cde=300 AND no_pdt=30;
```

Vérifier que la ligne de commande a été supprimée  
que le stock du produit est modifié (stock = 105)  
que le montant de la commande est modifié (diminution du montant)  
que les numéros de ligne de commande se suivent

**Remarque :**

1. Vous pouvez soit faire un trigger par événement, soit un seul trigger qui contient les traitements de chaque événement, je vous conseille un trigger par événement
2. pensez à utiliser les structures NEW et OLD

MODELE CONCEPTUEL DES DONNEESMODELE PHYSIQUE DES DONNEES