

Angelo Nichele

SAE 3.01

Marwan Al-Masri

Maxence Krill

Guychel Babella

Livrable

Triggers et Procédures

DISGUISE'HUB



Table des matières

1 Triggers.....	3
1.1 Code.....	3
1.2 Résultats.....	5
1.3 Tests.....	6
2 Procédures	8
2.1 Code.....	8
2.2 Résultats.....	9
2.3 Tests.....	10

1 Triggers

1.1 Code

```
CREATE TRIGGER check_qteStock
BEFORE INSERT ON Commander
FOR EACH ROW
BEGIN
    DECLARE v_qteProduit DECIMAL(4);

    SELECT qteProduit INTO v_qteProduit FROM Produit WHERE refProduit = NEW.refPro-
duit;

    IF NEW.qteCommandee <= 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La quantité commandée doit être supérieure à 0.';
    ELSE
        IF NEW.qteCommandee > v_qteProduit THEN
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La quantité commandée ne
peut pas être supérieure à la quantité en stock.';
        END IF;
    END IF;
END;
//

CREATE TRIGGER maj_qteProduit
AFTER INSERT ON Commander
FOR EACH ROW
BEGIN
    DECLARE v_qteActuelle DECIMAL(4);

    SELECT qteProduit INTO v_qteActuelle
    FROM Produit
    WHERE refProduit = NEW.refProduit;

    UPDATE Produit
    SET qteProduit = v_qteActuelle - NEW.qteCommandee
    WHERE refProduit = NEW.refProduit;
END;
//
```

```

CREATE TRIGGER check_dateExpCB
BEFORE INSERT ON Cartebleue
FOR EACH ROW
BEGIN
    IF NEW.dateExpCB <> DATE_FORMAT(NEW.dateExpCB, '%Y-%m-%d') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La date d'expiration de la carte bleue doit être au
format YYYY/MM/DD.';
    END IF;
END;
//

CREATE TRIGGER check_dateNaissanceClient
BEFORE INSERT ON Client
FOR EACH ROW
BEGIN
    IF NEW.dateNaissanceClient <> DATE_FORMAT(NEW.dateNaissanceClient, '%Y-%m-%d')
THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La date de naissance du client doit être au format
YYYY/MM/DD.';
    END IF;
END;
//

CREATE TRIGGER check_tailleProduit
BEFORE INSERT ON Produit
FOR EACH ROW
BEGIN
    IF NEW.tailleProduit NOT IN ('XS', 'S', 'M', 'L', 'XL', 'XXL') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La taille du produit doit être XS, S, M, L, XL ou
XXL.';
    END IF;
END;
//

```

1.2 Résultats

Déclencheurs 								
	Nom	Table	Action			Temps	Évènement	
<input type="checkbox"/>	check_dateCommande	Commande	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	check_dateExpCB	Cartebleue	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	check_dateNaissanceClient	Client	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	check_note	Avis	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	check_qteStock	Commander	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	check_tailleFR_and_tailleACommander	GuideTaille	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	check_tailleProduit	Produit	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	check_tailleSelonAgeEtSexe	CategTaille	 Éditer	 Exporter	 Supprimer	BEFORE	INSERT	
<input type="checkbox"/>	maj_qteProduit	Commander	 Éditer	 Exporter	 Supprimer	AFTER	INSERT	

1.3 Tests

Avant une insertion dans la table Commander :

SELECT * FROM Produit

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> | ☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

	refProduit	idCategorie	nomProduit	descProduit	prixProduit	qteProduit	tailleProduit	couleurProduit
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	100000	8	Pere Noël	Déguisement du pere Noël	50.99	50	S	rouge et blanc

```
INSERT INTO Commander (refProduit, idCommande, qteCommandee) VALUES  
(100000, 1, 2);
```

Après une insertion valide dans la table Commander :

SELECT * FROM Produit

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> | ☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

	refProduit	idCategorie	nomProduit	descProduit	prixProduit	qteProduit	tailleProduit	couleurProduit
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	100000	8	Pere Noël	Déguisement du pere Noël	50.99	48	S	rouge et blanc

Après une insertion non valide dans la table Commander :

```
INSERT INTO Commander (refProduit, idCommande, qteCommandee) VALUES  
(100000, 1, 0);
```

```
INSERT INTO Commander (refProduit, idCommande, qteCommandee) VALUES  
(100000, 1, 0);
```

MySQL a répondu : ?

#1644 - La quantité commandée doit être supérieure à 0.

```
INSERT INTO Commander (refProduit, idCommande, qteCommandee) VALUES  
(100000, 1, 51);
```

```
INSERT INTO Commander (refProduit, idCommande, qteCommandee) VALUES  
(100000, 1, 51);
```

MySQL a répondu : ?

#1644 - La quantité commandée ne peut pas être supérieure à la quantité en stock.

Après une insertion non valide dans la table Produit :

```
-- Insertion des produits  
INSERT INTO Produit (idCategorie, nomProduit, descProduit, prixProduit, qteProduit,  
tailleProduit, couleurProduit) VALUES  
(8, 'Pere Noël' , 'Déguisement du pere Noël', 50.99, 50, 'P', 'rouge et blanc');
```

```
-- Insertion des produits  
INSERT INTO Produit (idCategorie, nomProduit, descProduit, prixProduit, qteProduit, tailleProduit, couleurProduit) VALUES  
(8, 'Pere Noël' , 'Déguisement du pere Noël', 50.99, 50, 'P', 'rouge et blanc');
```

MySQL a répondu : ?

#1644 - La taille du produit doit être XS, S, M, L, XL ou XXL.

2 Procédures

2.1 Code

```
CREATE PROCEDURE PasserCommande(
    IN p_idClient INT,
    IN p_refProduit INT,
    IN p_qteCommandee INT,
    IN p_idPaielement INT,
    IN p_adrLivraison TEXT,
    IN p_codePostalLivraison DECIMAL(5),
    IN p_fraisLivraison DECIMAL(6,2)
)
BEGIN
    DECLARE v_qteProduit DECIMAL(4);

    SELECT qteProduit INTO v_qteProduit
    FROM Produit
    WHERE refProduit = p_refProduit;

    IF v_qteProduit IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Le produit spécifié n''existe
pas.';
    ELSE
        IF p_qteCommandee <= 0 THEN
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La quantité commandée doit
être supérieure à 0.';
        ELSE
            IF p_qteCommandee > v_qteProduit THEN
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La quantité commandée
ne peut pas être supérieure à la quantité en stock.';
            ELSE
                INSERT INTO Commande (idClient, idPaielement, dateCommande, frai-
sLivraison, adrLivraison, codePostalLivraison, statutCommande)
                VALUES (p_idClient, p_idPaielement, NOW(), p_fraisLivraison,
p_adrLivraison, p_codePostalLivraison, 'En cours de préparation');

                SET @lastCommandeId = LAST_INSERT_ID();

                INSERT INTO Commander (refProduit, idCommande, qteCommandee)
                VALUES (p_refProduit, @lastCommandeId, p_qteCommandee);

            END IF;
        END IF;
    END IF;
END;
//
```



```

CREATE PROCEDURE MettreAJourPrixProduit(
    IN p_refProduit INT,
    IN p_nouveauPrix DECIMAL(6,2)
)
BEGIN
    DECLARE v_qteProduit DECIMAL(4);

    SELECT qteProduit INTO v_qteProduit
    FROM Produit
    WHERE refProduit = p_refProduit;

    IF v_qteProduit IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Le produit spécifié n''existe
pas.';
    ELSE
        IF p_nouveauPrix <= 0 THEN
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Le nouveau prix doit être
supérieur à 0.';
        ELSE
            UPDATE Produit
            SET prixProduit = p_nouveauPrix
            WHERE refProduit = p_refProduit;
        END IF;
    END IF;
END;
//

```

2.2 Résultats

Procédures stockées ?

	Nom	Action				Type	Retourne
<input type="checkbox"/>	MettreAJourPrixProduit	Éditer	Exécuter	Exporter	Supprimer	PROCEDURE	
<input type="checkbox"/>	PasserCommande	Éditer	Exécuter	Exporter	Supprimer	PROCEDURE	

2.3 Tests

Exécution de la procédure « **PasserCommande** » :

Exécuter la procédure 'PasserCommande' ✕

Paramètres de procédure

Nom	Type	Fonction	Valeur
p_idClient	INT	<input type="text"/>	1
p_refProduit	INT	<input type="text"/>	100019
p_qteCommandee	INT	<input type="text"/>	4
p_idPaielement	INT	<input type="text"/>	1
p_adrLivraison	TEXT	<input type="text"/>	1 Rue des Fleurs
p_codePostalLivraison	DECIMAL	<input type="text"/>	13001
p_fraisLivraison	DECIMAL	<input type="text"/>	5.99

Exécuter Fermer

Mise à jour du stock dans la table Produit (Trigger maj_qteProduit) :

refProduit	idCategorie	nomProduit	descProduit	prixProduit	qteProduit	tailleProduit	couleurProduit
100019	12	Batman	Déguisement de Batman	55.99	96	XS	noir

Insertion dans la table Commande et Commander :

idCommande	idClient	idPaielement	dateCommande	fraisLivraison	adrLivraison	codePostalLivraison	paysCommande	statutC
106	1	1	2023-12-15	5.99	1 Rue des Fleurs	13001	NULL	En cou prépar

refProduit	idCommande	qteCommandee
100019	106	4

Exécution de la procédure « **MiseAJourPrixProduit** » :

Exécuter la procédure 'MettreAJourPrixProduit'

Paramètres de procédure

Nom	Type	Fonction	Valeur
p_refProduit	INT	<input type="text"/>	100000
p_nouveauPrix	DECIMAL	<input type="text"/>	78.99

Exécuter

Fermer

UPDATE dans la table Produit :

refProduit	idCategorie	nomProduit	descProduit	prixProduit	qteProduit	tailleProduit	couleurProduit
100000	8	Pere Noël	Déguisement du pere Noël	50.99	50	S	rouge et blanc

refProduit	idCategorie	nomProduit	descProduit	prixProduit	qteProduit	tailleProduit	couleurProduit
100000	8	Pere Noël	Déguisement du pere Noël	78.99	50	S	rouge et blanc