

# Compte Rendu - Procédures Stockées

## Sommaire

<b>1. Modification de la Base de Données</b>	<b>3</b>
Création d'une table <b>VARIETE</b>	3
Modification des tables <b>ENREGISTRER</b> , <b>COMMANDER</b> et <b>REGROUPER</b>	6
Ajout d'un attribut à la table <b>COMMANDE</b>	7
Diagramme de Classes de la Base de Données modifiée	7
<b>2. Procédures stockées</b>	<b>8</b>
<b>2.1 Procédure de traitement d'une nouvelle commande</b>	<b>8</b>
Spécifications techniques	8
Code de la procédure	9
Test de la procédure	13
<i><b>Cas nominaux</b></i>	13
PROGRAMME DE TEST	13
TRACE D'EXÉCUTION	17
<i><b>Cas d'erreur</b></i>	22
CLASSIFICATION DES CODES D'ERREUR	22
PROGRAMME DE TEST	23
RÉSULTAT DES TESTS	25
<b>2.2 Procédures de gestion des regroupements</b>	<b>26</b>
<b>2.2.1 Procédure d'ajout d'un regroupement</b>	<b>26</b>
Spécifications techniques	26
Code de la procédure	26
Test de la procédure	27
Cas nominal	27
Cas d'erreur	28
<b>2.2.2 Procédure de suppression d'un regroupement</b>	<b>29</b>
Spécifications techniques	29
Code de la procédure	29
Test de la procédure	30
Cas nominal	30
Cas d'erreur	30

# Procédures Stockées

*Vendredi 20 décembre 2024*



**Équipe 2B12**

Nolhan Biblocque

Léo Guinvarc'h

Victor Jockin

Mathys Laguilliez

Mucahit Lekesiz

**À destination de**

Laurent Nonne

**Formation**

BUT Informatique

2ème Année

Promotion 2024-2025

**Établissement**

IUT de Blagnac

Université Toulouse II – Jean Jaurès (31)

# 1. Modification de la Base de Données

## Création d'une table **VARIETE**

La table **VARIETE** a pour but de stocker des **variantes** de produits :

- Une variante est obligatoirement associée à un unique produit
- Un produit peut posséder plusieurs variantes
- Un produit est désormais simplement défini par un nom, une description, et une catégorie
- Une variante est définie par un produit, une **spécificité**, un prix et une quantité en stock
- La spécificité d'une variante peut correspondre à :
  - Une taille
  - Une couleur
  - Une matière
  - Une autonomie (ex : drones)
  - ...

### DICTIONNAIRE DES DONNÉES DE LA TABLE **VARIETE**

ATTRIBUT	DESCRIPTION	TYPE	CONTRAINTE
idVariete	Numéro de la variante	Entier	-
specVariete	Spécificité de la variante	Chaîne de 30 caractères	-
prixVariete	Prix de la variante	Nombre décimal à 22 chiffres dont 2 après la virgule	> 0.0
qteStockVariete	Quantité en stock de la variante	Nombre décimal à 10 chiffres	-

### REQUÊTE SQL DE CRÉATION DE LA TABLE

```
DROP TABLE IF EXISTS VARIETE ;
CREATE TABLE VARIETE(
    idVariete      INT NOT NULL,
    idProduit      INT NOT NULL,
    specVariete    VARCHAR(30)    DEFAULT NULL,
    prixVariete    DECIMAL(22,2)  NOT NULL,
    qteStockVariete DECIMAL(10,0)  NOT NULL,
    PRIMARY KEY (idVariete),
    FOREIGN KEY (idProduit) REFERENCES PRODUIT(idProduit)
) ;
```

## REQUÊTE SQL D'INSERTION DES DONNÉES

```
INSERT INTO VARIETE (idVariete, idProduit, specVariete, prixVariete, qteStockVariete)
VALUES
(101, 1, '10ML', '20.00', '55'),
(102, 1, '20ML', '37.50', '55'),
(201, 2, '10ML', '20.00', '55'),
(202, 2, '20ML', '37.50', '55'),
(301, 3, 'Rouge', '125.00', '40'),
(302, 3, 'Bleu', '125.00', '40'),
(303, 3, 'Rose', '125.00', '40'),
(401, 4, '1M', '40.00', '25'),
(402, 4, '1.5M', '60.00', '25'),
(501, 5, 'Argent', '190.00', '10'),
(502, 5, 'Noir', '190.00', '10'),
(503, 5, 'Rouge', '190.00', '10'),
(601, 6, '38', '215.00', '15'),
(602, 6, '39', '215.00', '15'),
(603, 6, '40', '215.00', '15'),
(604, 6, '41', '215.00', '15'),
(605, 6, '42', '215.00', '15'),
(606, 6, '43', '215.00', '15'),
(607, 6, '44', '215.00', '15'),
(608, 6, '45', '215.00', '15'),
(609, 6, '46', '215.00', '15'),
(701, 7, 'Saphir', '160.00', '10'),
(702, 7, 'Rubis', '160.00', '10'),
(703, 7, 'Diamant Synthétique', '160.00', '10'),
(801, 8, NULL, '55.00', '35'),
(901, 9, 'Lot de 5', '15.00', '20'),
(902, 9, 'Lot de 10', '25.00', '18'),
(903, 9, 'Lot de 20', '45.00', '15'),
(1001, 10, NULL, '60.00', '50'),
(1101, 11, '1m50', '70.00', '20'),
(1102, 11, '2m', '90.00', '15'),
(1103, 11, '3m', '130.00', '15'),
(1201, 12, 'Lot de 15', '50.00', '20'),
(1202, 12, 'Lot de 20', '85.00', '20'),
(1301, 13, NULL, '120.00', '30'),
(1401, 14, 'Noir', '115.00', '30'),
(1402, 14, 'Blanc', '115.00', '30'),
(1501, 15, 'Bleu', '560.00', '10'),
(1502, 15, 'Rouge', '560.00', '10'),
(1503, 15, 'Vert', '560.00', '10'),
(1504, 15, 'Jaune', '560.00', '10'),
(1601, 16, 'Autonomie 20h', '400000.00', '15'),
(1602, 16, 'Autonomie 50h', '1000000.00', '10'),
(1603, 16, 'Autonomie 150h', '2800000.00', '3'),
(1701, 17, '300cm3', '5000.00', '30'),
(1702, 17, '600cm3', '10000.00', '25'),
```

(1703, 17, '1100cm3', '15000.00', '15'),  
 (1801, 18, '1 place', '120000.00', '15'),  
 (1802, 18, '3 places', '250000.00', '10'),  
 (1901, 19, '38', '150.00', '20'),  
 (1902, 19, '39', '150.00', '20'),  
 (1903, 19, '40', '150.00', '20'),  
 (1904, 19, '41', '150.00', '20'),  
 (1905, 19, '42', '150.00', '20'),  
 (1906, 19, '43', '150.00', '20'),  
 (1907, 19, '44', '150.00', '20'),  
 (1908, 19, '45', '150.00', '20'),  
 (1909, 19, '46', '150.00', '20'),  
 (2001, 20, '38', '350.00', '20'),  
 (2002, 20, '39', '350.00', '20'),  
 (2003, 20, '40', '350.00', '20'),  
 (2004, 20, '41', '350.00', '20'),  
 (2005, 20, '42', '350.00', '20'),  
 (2006, 20, '43', '350.00', '20'),  
 (2007, 20, '44', '350.00', '20'),  
 (2008, 20, '45', '350.00', '20'),  
 (2009, 20, '46', '350.00', '20'),  
 (2101, 21, '38', '160.00', '15'),  
 (2102, 21, '39', '160.00', '15'),  
 (2103, 21, '40', '160.00', '15'),  
 (2104, 21, '41', '160.00', '15'),  
 (2105, 21, '42', '160.00', '15'),  
 (2106, 21, '43', '160.00', '15'),  
 (2107, 21, '44', '160.00', '15'),  
 (2108, 21, '45', '160.00', '15'),  
 (2109, 21, '46', '160.00', '15'),  
 (2201, 22, 'XS', '550.00', '10'),  
 (2202, 22, 'S', '550.00', '10'),  
 (2203, 22, 'M', '550.00', '10'),  
 (2204, 22, 'L', '550.00', '10'),  
 (2205, 22, 'XL', '550.00', '10'),  
 (2206, 22, 'XXL', '550.00', '10'),  
 (2301, 23, NULL, '890.00', '15'),  
 (2401, 24, 'XS', '135.00', '10'),  
 (2402, 24, 'S', '135.00', '10'),  
 (2403, 24, 'M', '135.00', '10'),  
 (2404, 24, 'L', '135.00', '10'),  
 (2405, 24, 'XL', '135.00', '10'),  
 (2501, 25, 'Lot de 3', '500.00', '20'),  
 (2502, 25, 'Lot de 5', '800.00', '15'),  
 (2503, 25, 'Lot de 10', '1500.00', '15'),  
 (2601, 26, 'Verres polarisés', '180.00', '25'),  
 (2602, 26, 'Verres non polarisés', '180.00', '25'),  
 (2701, 27, 'Rouge', '350.00', '10'),  
 (2702, 27, 'Noir', '350.00', '10'),  
 (2703, 27, 'Gris', '350.00', '10'),  
 (2801, 28, 'Design 1', '60.00', '50'),

```
(2802, 28, 'Design 2', '60.00', '50'),
(2803, 28, 'Design 3', '60.00', '50'),
(2901, 29, 'Rouge', '45.00', '20'),
(2902, 29, 'Bleu', '45.00', '20'),
(2903, 29, 'Mauve', '45.00', '20'),
(3001, 30, NULL, '220.00', '20'),
(3101, 31, NULL, '35.00', '30'),
(3201, 32, NULL, '75.00', '30'),
(3301, 33, 'Standard', '50.00', '40'),
(3302, 33, 'Avancée', '100.00', '25'),
(3401, 34, 'Petit', '15.00', '40'),
(3402, 34, 'Moyen', '25.00', '40'),
(3403, 34, 'Grand', '35.00', '30'),
(3501, 35, 'Comprimé (lot de 5)', '30.00', '20'),
(3502, 35, 'Liquide (lot de 3)', '35.00', '25'),
(3601, 36, 'Petit (pour zones ciblées)', '75.00', '35'),
(3602, 36, 'Grand (pour tout le corps)', '120.00', '15') ;
```

## Modification des tables **ENREGISTRER**, **COMMANDER** et **REGROUPER**

### MODIFICATION DE LA TABLE **ENREGISTRER**

```
-- changement du nom de l'attribut `idProduit` en `idVariete`
ALTER TABLE ENREGISTRER CHANGE COLUMN idProduit idVariete INT NOT NULL ;

-- suppression de l'ancienne contrainte de clé étrangère sur `idProduit`
ALTER TABLE ENREGISTRER DROP FOREIGN KEY fk_idProduit_enregistrer_produit ;

-- ajout de la nouvelle contrainte de clé étrangère sur `idVariete`
ALTER TABLE ENREGISTRER ADD FOREIGN KEY (idVariete) REFERENCES VARIETE(idVariete) ;
```

### MODIFICATION DE LA TABLE **COMMANDER**

```
-- changement du nom de l'attribut `idProduit` en `idVariete`
ALTER TABLE COMMANDER CHANGE COLUMN idProduit idVariete INT NOT NULL ;

-- suppression de l'ancienne contrainte de clé étrangère sur `idProduit`
ALTER TABLE COMMANDER DROP FOREIGN KEY fk_idProduit_commander_produit ;

-- ajout de la nouvelle contrainte de clé étrangère sur `idVariete`
ALTER TABLE COMMANDER ADD FOREIGN KEY (idVariete) REFERENCES VARIETE(idVariete) ;
```

## MODIFICATION DE LA TABLE REGROUPER

```
-- changement du nom de l'attribut 'idProduit' en 'idVariete'
ALTER TABLE REGROUPER CHANGE COLUMN idProduit idVariete INT NOT NULL ;

-- suppression de l'ancienne contrainte de clé étrangère sur 'idProduit'
ALTER TABLE REGROUPER DROP FOREIGN KEY fk_idProduit_regrouper_produit ;

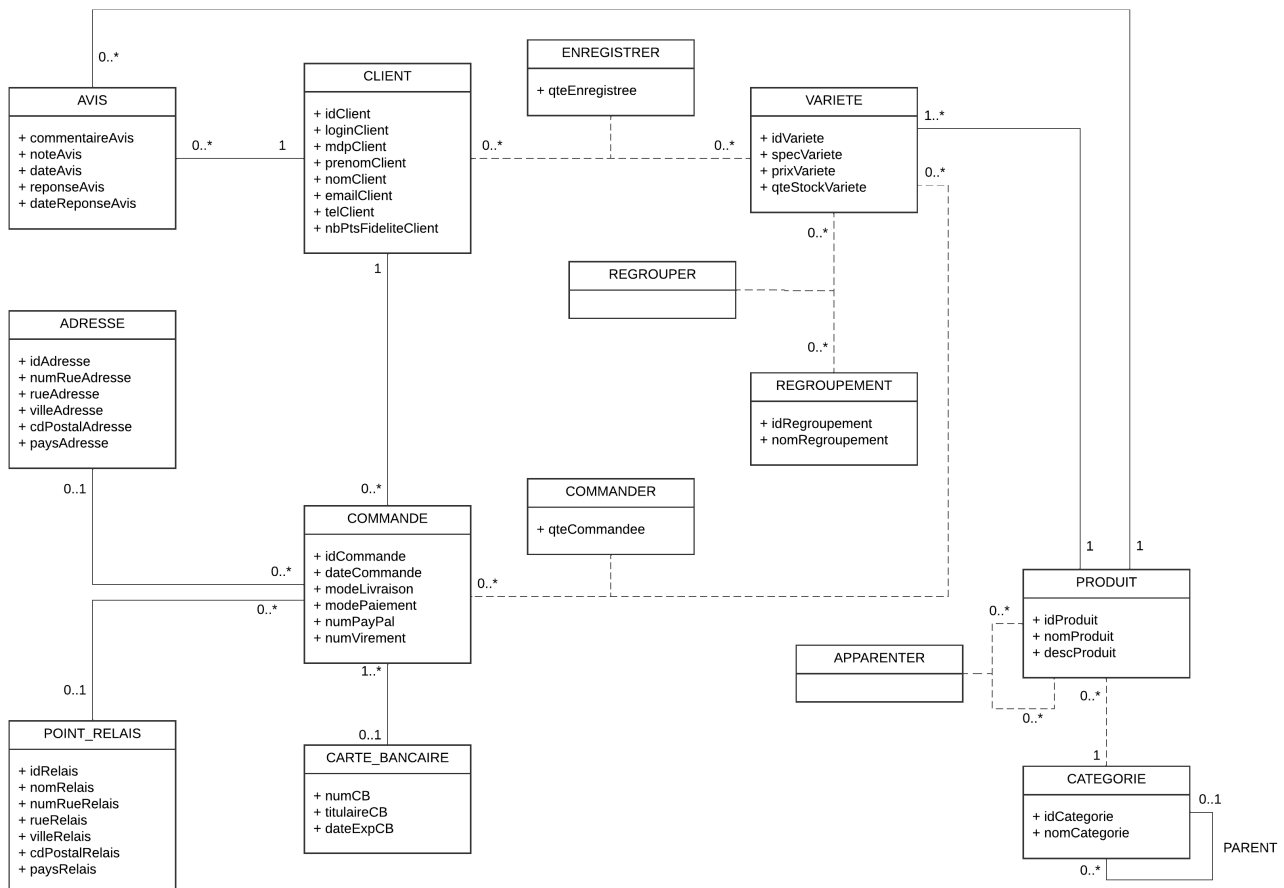
-- ajout de la nouvelle contrainte de clé étrangère sur 'idVariete'
ALTER TABLE REGROUPER ADD FOREIGN KEY (idVariete) REFERENCES VARIETE(idVariete) ;
```

## Ajout d'un attribut à la table **COMMANDE**

## REQUÊTE SQL

```
-- ajout d'un attribut décrivant le montant total de la commande
ALTER TABLE COMMANDE
ADD COLUMN montantCommande DECIMAL(22,2) DEFAULT 0 ;
```

## Diagramme de Classes de la Base de Données modifiée



## 2. Procédures stockées

### 2.1 Procédure de traitement d'une nouvelle commande

#### Spécifications techniques

La procédure SQL/PSM `passerCommande` répond aux spécifications suivantes :

##### *Entrées :*

- Numéro (ID) du client passant la commande
- Mode de livraison de la commande
- Numéro (ID) de l'adresse de livraison de la commande (domicile ou point relais)
- Mode de paiement de la commande
- Éventuel numéro de carte bancaire utilisé pour le paiement de la commande
- Éventuel numéro PayPal utilisé pour le paiement de la commande
- Éventuel numéro de virement utilisé pour le paiement de la commande

##### *Opérations réalisées :*

1. Récupération des produits enregistrés dans le panier du client passant la commande
2. Création d'un objet de type `COMMANDE` à partir des paramètres donnés en entrée
3. Ajout des produits récupérés à la commande
4. Nettoyage du panier du client (retrait des produits commandés)
5. Validation des transactions effectuées (`COMMIT`)

##### *Cas d'erreurs traités :*

- Client inconnu ou non renseigné
- Mode de livraison invalide ou non renseigné
- Adresse de livraison inconnue (ne correspondant à aucun domicile / point relais enregistré) ou non renseignée
- Mode de paiement invalide ou non renseigné
- Carte bancaire inconnue ou non renseignée
- Numéro PayPal non renseigné
- Numéro de virement non renseigné



## Code de la procédure

### PROCÉDURE SQL/PSM

```
/*
 * Procédure de traitement d'une nouvelle commande.
 *
 * Date de dernière modification :
 * - Lundi 23 décembre 2024 -
 *
 * Auteur : Victor Jockin
 * - Équipe 2B12 -
 */

-- paramétrage du délimiteur
DELIMITER /

-- suppression de la procédure si existante
DROP PROCEDURE IF EXISTS passerCommande /

-- définition de la procédure
CREATE PROCEDURE passerCommande(
    pIdClient INT,
    pModeLivraison VARCHAR(15),
    pIdAdresseDeLivraison INT,
    pModePaiement VARCHAR(10),
    pNumCB CHAR(16),
    pNumPaypal DECIMAL(10),
    pNumVirement DECIMAL(10)
)
BEGIN
    -- déclaration des variables
    -- -----
    -- variables relatives à la commande à créer
    DECLARE varDateCommande DATE ;
    DECLARE varIdAdresse INT ;
    DECLARE varIdRelais INT ;
    DECLARE varNumCB CHAR(16) ;
    DECLARE varNumPaypal DECIMAL(10) ;
    DECLARE varNumVirement DECIMAL(10) ;
    DECLARE varMontant DECIMAL(22,2) ;
    -- variables relatives aux produits commandés
    DECLARE varIdCommande INT ;
    DECLARE varIdVProduit INT ;
    DECLARE varPrixVProduit DECIMAL(22,0) ;
    DECLARE varQteEnregistree DECIMAL(10,0) ;
    DECLARE varQteStockVariete DECIMAL(10,0) ;

    -- date de la commande
    -- ID d'une adresse client
    -- ID d'un point relais
    -- numéro de carte bancaire
    -- numéro de paiement PayPal
    -- numéro de virement
    -- montant total de la commande
    -- ID de la commande insérée
    -- ID d'une variante de produit
    -- prix d'une variante de produit
    -- quantité enregistrée pour une
    -- variante de produit
    -- quantité en stock d'une
    -- variante de produit
```

```

DECLARE isFinCurseur BOOLEAN DEFAULT FALSE ;    -- indique si le curseur a été
                                                    -- entièrement lu

-- compteur de lignes pour exceptions
DECLARE varCpt INT ;
-- curseur de récupération des variantes de produits commandés
DECLARE curseurVProduitsCommandees CURSOR FOR
    SELECT E.idVariete, E.qteEnregistree, V.prixVariete
    FROM ENREGISTRER E INNER JOIN VARIETE V
        ON E.idVariete = V.idVariete
    WHERE E.idClient = pIdClient ;
-- gestionnaire d'exception pour lecture du curseur
DECLARE CONTINUE HANDLER FOR NOT FOUND SET isFinCurseur = TRUE ;

-- initialisation des variables relatives à la commande
SET varDateCommande = CURDATE() ;
SET varIdAdresse     = NULL ;
SET varIdRelais       = NULL ;
SET varNumCB          = NULL ;
SET varNumPaypal      = NULL ;
SET varNumVirement   = NULL ;
SET varMontant        = 0 ;

-- traitement des exceptions
-- -----
-- vérification de la validité de l'ID Client
IF pIdClient IS NULL THEN
    SIGNAL SQLSTATE '45010'
    SET MESSAGE_TEXT = 'Erreur : Client non renseigné.',
        MYSQL_ERRNO = 45010 ;
END IF ;
SELECT COUNT(*) INTO varCpt FROM CLIENT WHERE idClient = pIdClient ;
IF varCpt != 1 THEN
    SIGNAL SQLSTATE '45011'
    SET MESSAGE_TEXT = 'Erreur : Client inconnu.',
        MYSQL_ERRNO = 45011 ;
END IF ;
-- traitement du mode de livraison
IF pModeLivraison IS NULL THEN
    SIGNAL SQLSTATE '45020'
    SET MESSAGE_TEXT = 'Erreur : Mode de livraison non renseigné.',
        MYSQL_ERRNO = 45020 ;
END IF ;
IF pIdAdresseDeLivraison IS NULL THEN
    SIGNAL SQLSTATE '45030'
    SET MESSAGE_TEXT = 'Erreur : Adresse de livraison non renseignée.',
        MYSQL_ERRNO = 45030 ;
END IF ;
CASE WHEN pModeLivraison = 'DOMICILE' THEN
    SELECT COUNT(*) INTO varCpt
    FROM ADRESSE WHERE idAdresse = pIdAdresseDeLivraison ;
    IF varCpt != 1 THEN

```

```

        SIGNAL SQLSTATE '45031'
        SET MESSAGE_TEXT = 'Erreur : Adresse ne correspondant à aucun domicile
                               enregistré.',
        MYSQL_ERRNO = 45031 ;
    END IF ;
    SET varIdAdresse = pIdAdresseDeLivraison ;
    WHEN pModeLivraison = 'RELAIS' THEN
        SELECT COUNT(*) INTO varCpt
        FROM POINT_RELAIS WHERE idRelais = pIdAdresseDeLivraison ;
        IF varCpt != 1 THEN
            SIGNAL SQLSTATE '45032'
            SET MESSAGE_TEXT = 'Erreur : Adresse ne correspondant à aucun point
                               relais enregistré.',
            MYSQL_ERRNO = 45032 ;
        END IF ;
        SET varIdRelais = pIdAdresseDeLivraison ;
    ELSE
        SIGNAL SQLSTATE '45021'
        SET MESSAGE_TEXT = 'Erreur : Mode de livraison invalide.',
        MYSQL_ERRNO = 45021 ;
    END CASE ;
    -- traitement du mode de paiement
    IF pModePaiement IS NULL THEN
        SIGNAL SQLSTATE '45040'
        SET MESSAGE_TEXT = 'Erreur : Mode de paiement non renseigné.',
        MYSQL_ERRNO = 45040 ;
    END IF ;
    CASE WHEN pModePaiement = 'CB' THEN
        IF pNumCB IS NULL THEN
            SIGNAL SQLSTATE '45050'
            SET MESSAGE_TEXT = 'Erreur : Carte bancaire non renseignée.',
            MYSQL_ERRNO = 45050 ;
        END IF ;
        SELECT COUNT(*) INTO varCpt FROM CARTE_BANCAIRE WHERE numCB = pNumCB ;
        IF varCpt != 1 THEN
            SIGNAL SQLSTATE '45051'
            SET MESSAGE_TEXT = 'Erreur : Carte bancaire inconnue.',
            MYSQL_ERRNO = 45051 ;
        END IF ;
        SET varNumCB = pNumCB ;
    WHEN pModePaiement = 'PAYPAL' THEN
        IF pNumPaypal IS NULL THEN
            SIGNAL SQLSTATE '45060'
            SET MESSAGE_TEXT = 'Erreur : Numéro PayPal non renseigné.',
            MYSQL_ERRNO = 45060 ;
        END IF ;
        SET varNumPaypal = pNumPaypal ;
    WHEN pModePaiement = 'VIREMENT' THEN
        IF pNumVirement IS NULL THEN
            SIGNAL SQLSTATE '45070'
            SET MESSAGE_TEXT = 'Erreur : Numéro de virement non renseigné.',

```

```

        MYSQL_ERRNO = 45070 ;
    END IF ;
    SET varNumVirement = pNumVirement ;
ELSE
    SIGNAL SQLSTATE '45041'
    SET MESSAGE_TEXT = 'Erreur : Mode de paiement invalide.',
        MYSQL_ERRNO = 45041 ;
END CASE ;

-- traitement de la commande
-- -----
-- insertion de la commande
INSERT INTO COMMANDE (idClient, dateCommande, modeLivraison, idAdresse, idRelais,
                    modePaiement, numCB, numPaypal, numVirement)
VALUES (pIdClient, varDateCommande, pModeLivraison, varIdAdresse, varIdRelais,
        pModePaiement, varNumCB, varNumPaypal, varNumVirement) ;
-- récupération de l'ID de la commande
SET varIdCommande = LAST_INSERT_ID() ;
-- insertion des variantes de produits commandés
OPEN curseurVProduitsCommandees ;
BOUCLE: LOOP
    FETCH curseurVProduitsCommandees
    INTO varIdVProduit, varQteEnregistree, varPrixVProduit ;
    IF isFinCurseur THEN LEAVE BOUCLE ; END IF ;
    -- mise à jour de la quantité commandée en cas de dépassement
    -- du stock disponible
    SELECT qteStockVariete INTO varQteStockVariete
    FROM VARIETE WHERE idVariete = varIdVProduit ;
    IF varQteEnregistree > varQteStockVariete THEN
        SET varQteEnregistree = varQteStockVariete ;
    END IF ;
    -- ajout de la variante du produit à la commande
    INSERT INTO COMMANDER (idCommande, idVariete, qteCommandee)
    VALUES (varIdCommande, varIdVProduit, varQteEnregistree) ;
    -- mise à jour du stock de la variante de produit
    UPDATE VARIETE
    SET qteStockVariete = qteStockVariete - varQteEnregistree
    WHERE idVariete = varIdVProduit ;
    -- calcul du montant de la commande
    SET varMontant = varMontant + varQteEnregistree * varPrixVProduit ;
END LOOP ;
CLOSE curseurVProduitsCommandees ;
-- mise à jour du montant de la commande
UPDATE COMMANDE SET montantCommande = varMontant WHERE idCommande = varIdCommande ;
-- nettoyage du panier (suppression des produits commandés)
DELETE FROM ENREGISTRER WHERE idClient = pIdClient ;
-- validation des transactions
COMMIT ;
END /
DELIMITER ;

```

## Test de la procédure

### Cas nominaux

#### PROGRAMME DE TEST

##### MISE EN PLACE DES OBJETS DE TEST

```
-- mise en place des objets de test
-- -----

-- insertion de 4 clients fictifs
INSERT INTO CLIENT (prenomClient, nomClient, loginClient, mdpClient, telephoneClient,
                    emailClient)
VALUES ('Anu-Ujin', 'Ganpurev', 'missuji', '-', '0714141414', 'missuji@gmail.com') ;
SET @varIdClientTest1 = LAST_INSERT_ID() ;
INSERT INTO CLIENT (prenomClient, nomClient, loginClient, mdpClient, telephoneClient,
                    emailClient)
VALUES ('John', 'Powell', 'johnpowell', '-', '(555) 555-1234', 'jpowell@outlook.com');
SET @varIdClientTest2 = LAST_INSERT_ID() ;
INSERT INTO CLIENT (prenomClient, nomClient, loginClient, mdpClient, telephoneClient,
                    emailClient)
VALUES ('Azuly', 'Azuly', 'azuly', '-', '0724252425', 'aazzuullyy@gmail.com') ;
SET @varIdClientTest3 = LAST_INSERT_ID() ;
INSERT INTO CLIENT (prenomClient, nomClient, loginClient, mdpClient, telephoneClient,
                    emailClient)
VALUES ('Marie', 'Chouvrey', 'morito415', '-', '0715151515', 'morito@gmail.com') ;
SET @varIdClientTest4 = LAST_INSERT_ID() ;

-- insertion de 4 paniers fictifs (1 pour chaque client fictif)
INSERT INTO ENREGISTRER (idClient, qteEnregistree, idVariete)
VALUES (@varIdClientTest1, 1, 102),
        (@varIdClientTest1, 1, 501),
        (@varIdClientTest2, 1, 101),
        (@varIdClientTest2, 2, 202),
        (@varIdClientTest3, 3, 301),
        (@varIdClientTest3, 1, 401),
        (@varIdClientTest3, 1, 502),
        (@varIdClientTest4, 1, 1501) ;

-- insertion d'une carte bancaire fictive
INSERT INTO CARTE_BANCAIRE (numCB, dateExpCB, titulaireCB)
VALUES ('111122233334444', '01/27', 'ANU-UJIN GANPUREV') ;
```

## AFFICHAGES AVANT EXÉCUTION

```
-- affichages avant exécution
-- -----

-- affichage des paniers
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest1 ;
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest2 ;
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest3 ;
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest4 ;

-- affichage des commandes
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest1 ;
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest2 ;
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest3 ;
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest4 ;

-- affichage des variantes de produits commandées
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest1 ;
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest2 ;
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest3 ;
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest4 ;
```

## TEST DES CAS NOMINAUX

```
-- test des cas nominaux
-- -----

-- test [01] : commande livrée à domicile et payée par carte bancaire
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 227, 'CB', '1111222233334444',
                    NULL, NULL) ;

-- test [02] : commande livrée à domicile et payée par PayPal
CALL passerCommande(@varIdClientTest2, 'DOMICILE', 99, 'PAYPAL', NULL,
                    '1234567890', NULL) ;
```

```
-- test [03] : commande livrée en point relais et payée par virement bancaire
CALL passerCommande(@varIdClientTest3, 'RELAIS', 154, 'VIREMENT', NULL,
                    NULL, '1234567890') ;

-- test [04] : commande livrée à domicile et payée par carte bancaire avec
--            écrasement des paramètres non utilisés (mode de paiement)
CALL passerCommande(@varIdClientTest4, 'DOMICILE', 299, 'CB', '1111222233334444',
                    '1234567890', '1234567890') ;
```

## AFFICHAGES APRÈS EXÉCUTION

```
-- affichages après exécution
-- -----

-- affichage des paniers
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest1 ;
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest2 ;
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest3 ;
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest4 ;

-- affichage des commandes
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest1 ;
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest2 ;
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest3 ;
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest4 ;

-- affichage des variantes de produits commandées
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest1 ;
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest2 ;
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest3 ;
SELECT V.* FROM VARIETE V
INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete
INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande
WHERE CE.idClient = @varIdClientTest4 ;
```

## SUPPRESSION DES OBJETS DE TEST

```
-- suppression des objets de test
-- -----

-- suppression des éventuels paniers restants
DELETE FROM ENREGISTRER
WHERE idClient IN (
    @varIdClientTest1, @varIdClientTest2, @varIdClientTest3, @varIdClientTest4
) ;

-- suppression des variantes de produits commandées
DELETE FROM COMMANDER
WHERE idCommande IN (
    SELECT idCommande
    FROM COMMANDE
    WHERE idClient IN (
        @varIdClientTest1, @varIdClientTest2, @varIdClientTest3, @varIdClientTest4
    )
) ;

-- suppression des commandes
DELETE FROM COMMANDE
WHERE idClient IN (
    @varIdClientTest1, @varIdClientTest2, @varIdClientTest3, @varIdClientTest4
) ;

-- suppression de la carte bancaire fictive
DELETE FROM CARTE_BANCAIRE WHERE numCB = '1111222233334444' ;

-- suppression des clients fictifs
DELETE FROM CLIENT
WHERE idClient IN (
    @varIdClientTest1, @varIdClientTest2, @varIdClientTest3, @varIdClientTest4
) ;
```



## AFFICHAGES AVANT EXÉCUTION

## PANIER DES CLIENTS FICTIFS

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0.0020 seconde(s).)

```
-- affichages avant exécution SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest1;
```

[\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Créer le code source PHP \]](#)☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

				idClient	idVariete	qteEnregistree
<input type="checkbox"/>	Éditer	Copier	Supprimer	778	102	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	778	501	1

☐ Tout cocher | Avec la sélection : Éditer Copier Supprimer Exporter

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0.0023 seconde(s).)

```
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest2;
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

				idClient	idVariete	qteEnregistree
<input type="checkbox"/>	Éditer	Copier	Supprimer	779	101	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	779	202	2

☐ Tout cocher | Avec la sélection : Éditer Copier Supprimer Exporter

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0.0019 seconde(s).)

```
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest3;
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

				idClient	idVariete	qteEnregistree
<input type="checkbox"/>	Éditer	Copier	Supprimer	780	301	3
<input type="checkbox"/>	Éditer	Copier	Supprimer	780	401	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	780	502	1

☐ Tout cocher | Avec la sélection : Éditer Copier Supprimer Exporter

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0.0019 seconde(s).)

```
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest4;
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

+ Options

				idClient	idVariete	qteEnregistree
<input type="checkbox"/>	Éditer	Copier	Supprimer	781	1501	1

☐ Tout cocher | Avec la sélection : Éditer Copier Supprimer Exporter

## COMMANDES DES CLIENTS FICTIFS

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0022 seconde(s).)

```
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest1;
```

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

idCommande	dateCommande	modeLivraison	modePaiement	numPaypal	numVirement	idClient	idAdresse
------------	--------------	---------------	--------------	-----------	-------------	----------	-----------

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0023 seconde(s).)

```
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest2;
```

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

idCommande	dateCommande	modeLivraison	modePaiement	numPaypal	numVirement	idClient	idAdresse
------------	--------------	---------------	--------------	-----------	-------------	----------	-----------

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0022 seconde(s).)

```
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest3;
```

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

idCommande	dateCommande	modeLivraison	modePaiement	numPaypal	numVirement	idClient	idAdresse
------------	--------------	---------------	--------------	-----------	-------------	----------	-----------

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0021 seconde(s).)

```
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest4;
```

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

idCommande	dateCommande	modeLivraison	modePaiement	numPaypal	numVirement	idClient	idAdresse
------------	--------------	---------------	--------------	-----------	-------------	----------	-----------

## VARIANTES DE PRODUITS COMMANDÉES PAR LES CLIENTS FICTIFS

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0060 seconde(s).)

```
SELECT V.* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest1;
```

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

idVariete	idProduit	specVariete	prixVariete	qteStockVariete
-----------	-----------	-------------	-------------	-----------------

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0087 seconde(s).)

```
SELECT V.* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest2;
```

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

idVariete	idProduit	specVariete	prixVariete	qteStockVariete
-----------	-----------	-------------	-------------	-----------------

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0089 seconde(s).)

```
SELECT V.* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest3;
```

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

idVariete	idProduit	specVariete	prixVariete	qteStockVariete
-----------	-----------	-------------	-------------	-----------------

✔ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0086 seconde(s).)

```
SELECT V.* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest4;
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)

idVariete idProduit specVariete

prixVariete qteStockVariete

## AFFICHAGES APRÈS EXÉCUTION

### PANIER DES CLIENTS FICTIFS

✔ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0026 seconde(s).)

```
non renseigné -- CALL passerCommande(@varIdClientTest1, NULL, 227, 'CB', '1111222233334444', NULL, NULL) ; -- mode de livraison
invalide -- CALL passerCommande(@varIdClientTest1, 'HIBOU', 227, 'CB', '1111222233334444', NULL, NULL) ; -- adresse de livraison non
renseignée -- CALL passerCommande(@varIdClientTest1, 'DOMICILE', NULL, 'CB', '1111222233334444', NULL, NULL) ; -- adresse de livraison
inconnue -- CALL passerCommande(@varIdClientTest1, 'DOMICILE', 154, 'CB', '1111222233334444', NULL, NULL) ; -- CALL
passerCommande(@varIdClientTest1, 'RELAIS', 227, 'CB', '1111222233334444', NULL, NULL) ; -- mode de paiement non renseigné -- CALL
passerCommande(@varIdClientTest1, 'DOMICILE', 227, NULL, '1111222233334444[...]
```

[\[ Éditer \]](#)

idClient idVariete qteEnregistree

✔ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0029 seconde(s).)

```
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest2;
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)

idClient idVariete qteEnregistree

✔ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0028 seconde(s).)

```
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest3;
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)

idClient idVariete qteEnregistree

✔ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0031 seconde(s).)

```
SELECT * FROM ENREGISTRER WHERE idClient = @varIdClientTest4;
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)

idClient idVariete qteEnregistree

### COMMANDES DES CLIENTS FICTIFS

✔ Affichage des lignes 0 - 0 (total de 1, traitement en 0.0034 seconde(s).) [idCommande: 401... - 401...]

```
SELECT * FROM COMMANDE WHERE idClient = @varIdClientTest1 ORDER BY `idCommande` DESC
```

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)

☐ Tout afficher

Nombre de lignes : 

25

Filter les lignes: 

Chercher dans cette tat

+ Options

	idCommande	dateCommande	modeLivraison	modePaiement	numPaypal	numVirement	idClient	idAdresse	idRelais	numC
<input type="checkbox"/> Éditer	401	2024-12-24	DOMICILE	CB	NULL	NULL	778	227	NULL	111122

↑

☐ Tout cocher

Avec la sélection :

Éditer

Copier

Supprimer

Exporter

Affichage des lignes 0 - 0 (total de 1, traitement en 0.0022 seconde(s).) [idCommande: 402... - 402...]

SELECT \* FROM COMMANDE WHERE idClient = @varIdClientTest2 ORDER BY `idCommande` DESC

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

Tout afficher

 | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette tak

+ Options

↩ ↪

idCommande dateCommande modeLivraison modePaiement numPaypal numVirement idClient idAdresse idRelais num

Éditer Copier Supprimer

402 2024-12-24 DOMICILE PAYPAL 1234567890 NULL 779 299 NULL NUI

↑

Tout cocher

 Avec la sélection : 

Éditer Copier Supprimer Exporter

Affichage des lignes 0 - 0 (total de 1, traitement en 0.0022 seconde(s).) [idCommande: 403... - 403...]

SELECT \* FROM COMMANDE WHERE idClient = @varIdClientTest3 ORDER BY `idCommande` DESC

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

Tout afficher

 | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette tak

+ Options

↩ ↪

idCommande dateCommande modeLivraison modePaiement numPaypal numVirement idClient idAdresse idRelais num

Éditer Copier Supprimer

403 2024-12-24 RELAIS VIREMENT NULL 1234567890 780 NULL 154 NUI

↑

Tout cocher

 Avec la sélection : 

Éditer Copier Supprimer Exporter

Affichage des lignes 0 - 0 (total de 1, traitement en 0.0022 seconde(s).) [idCommande: 404... - 404...]

SELECT \* FROM COMMANDE WHERE idClient = @varIdClientTest4 ORDER BY `idCommande` DESC

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

Tout afficher

 | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette tak

+ Options

↩ ↪

idCommande dateCommande modeLivraison modePaiement numPaypal numVirement idClient idAdresse idRelais nu

Éditer Copier Supprimer

404 2024-12-24 DOMICILE CB NULL NULL 781 299 NULL 111

↑

Tout cocher

 Avec la sélection : 

Éditer Copier Supprimer Exporter

VARIANTES DE PRODUITS COMMANDÉES PAR LES CLIENTS FICTIFS

Affichage des lignes 0 - 1 (total de 2, traitement en 0.0058 seconde(s).)

SELECT V.\* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest1 ORDER BY `idVariete` ASC

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

Tout afficher

 | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette tak | Trier par clé : PRIMARY (ASC)

+ Options

idVariete 1 idProduit specVariete prixVariete qteStockVariete

102 1 20ML 37.50 46

501 5 Argent 190.00 7

20

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0.0056 seconde(s).)

```
SELECT V.* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest2 ORDER BY `idVariete` ASC
```

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

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes:  Trier par clé : PRIMARY (ASC)

+ Options

idVariete	idProduit	specVariete	prixVariete	qteStockVariete
101	1	10ML	20.00	48
202	2	20ML	37.50	47

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0.0057 seconde(s).)

```
SELECT V.* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest3 ORDER BY `idVariete` ASC
```

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

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes:  Trier par clé : PRIMARY (ASC)

+ Options

idVariete	idProduit	specVariete	prixVariete	qteStockVariete
301	3	Rouge	125.00	37
401	4	1M	40.00	24
502	5	Noir	190.00	9

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0.0057 seconde(s).)

```
SELECT V.* FROM VARIETE V INNER JOIN COMMANDER CR ON V.idVariete = CR.idVariete INNER JOIN COMMANDE CE ON CR.idCommande = CE.idCommande WHERE CE.idClient = @varIdClientTest4 ORDER BY `idVariete` ASC
```

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

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes:

+ Options

idVariete	idProduit	specVariete	prixVariete	qteStockVariete
1501	15	Bleu	560.00	9

## ***Cas d'erreur***

### **CLASSIFICATION DES CODES D'ERREUR**

Cette section détaille l'ensemble des codes d'erreurs pouvant être retournés par la procédure [passerCommande](#).

#### **4501 : *Erreur sur client***

- 45010 : Client non renseigné
- 45011 : Client inconnu

#### **4502 : *Erreur sur mode de livraison***

- 45020 : Mode de livraison non renseigné
- 45021 : Mode de livraison invalide

#### **4503 : *Erreur sur adresse de livraison***

- 45030 : Adresse de livraison non renseignée
- 45031 : Adresse ne correspondant à aucun domicile enregistré
- 45032 : Adresse ne correspondant à aucun point relais enregistré

#### **4504 : *Erreur sur mode de paiement***

- 45040 : Mode de paiement non renseigné
- 45041 : Mode de paiement invalide

#### **4505 : *Erreur sur carte bancaire***

- 45050 : Carte bancaire non renseignée
- 45051 : Carte bancaire inconnue

#### **4506 : *Erreur sur numéro PayPal***

- 45060 : Numéro PayPal non renseigné

#### **4507 : *Erreur sur numéro de virement***

- 45070 : Numéro de virement non renseigné

## MISE EN PLACE DES OBJETS DE TEST

```
-- mise en place des objets de test
-- -----

-- insertion d'un client fictif
INSERT INTO CLIENT (prenomClient, nomClient, loginClient, mdpClient, telephoneClient,
                    emailClient)
VALUES ('Anu-Ujin', 'Ganpurev', 'missuji', '-', '0714141414', 'missuji@gmail.com') ;

-- insertion d'une carte bancaire fictive
INSERT INTO CARTE_BANCAIRE (numCB, dateExpCB, titulaireCB)
VALUES ('1111222233334444', '01/27', 'ANU-UJIN GANPUREV') ;
```

## TEST DES CAS D'ERREUR

```
-- test des cas d'erreur
-- -----

-- test [10] : client non renseigné
CALL passerCommande(NULL, 'DOMICILE', 227, 'CB', '1111222233334444', NULL, NULL) ;
-- test [11] : client inconnu
CALL passerCommande(1212, 'DOMICILE', 227, 'CB', '1111222233334444', NULL, NULL) ;

-- test [20] : mode de livraison non renseigné
CALL passerCommande(@varIdClientTest1, NULL, 227, 'CB', '1111222233334444',
                    NULL, NULL) ;
-- test [21] : mode de livraison invalide
CALL passerCommande(@varIdClientTest1, 'HIBOU', 227, 'CB', '1111222233334444',
                    NULL, NULL) ;

-- test [30] : adresse de livraison non renseignée
CALL passerCommande(@varIdClientTest1, 'DOMICILE', NULL, 'CB', '1111222233334444',
                    NULL, NULL) ;
-- test [31] : adresse de livraison inconnue (domicile inconnu)
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 154, 'CB', '1111222233334444',
                    NULL, NULL) ;
-- test [32] : adresse de livraison inconnue (point relais inconnu)
CALL passerCommande(@varIdClientTest1, 'RELAIS', 227, 'CB', '1111222233334444',
                    NULL, NULL) ;

-- test [40] : mode de paiement non renseigné
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 227, NULL, '1111222233334444',
                    NULL, NULL) ;
```

```
-- test [41] : mode de paiement invalide
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 227, 'LINGOT', '1111222233334444',
                    NULL, NULL) ;

-- test [50] : carte bancaire non renseignée
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 227, 'CB', NULL, NULL, NULL) ;
-- test [51] : carte bancaire inconnue
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 227, 'CB', '5555666677778888',
                    NULL, NULL) ;

-- test [60] : numéro PayPal non renseigné
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 227, 'PAYPAL', NULL, NULL, NULL) ;
-- test [70] : numéro de virement non renseigné
CALL passerCommande(@varIdClientTest1, 'DOMICILE', 227, 'VIREMENT', NULL, NULL, NULL);
```

## SUPPRESSION DES OBJETS DE TEST

```
-- suppression des objets de test
-- -----

-- suppression de la carte bancaire fictive
DELETE FROM CARTE_BANCAIRE WHERE numCB = '1111222233334444' ;

-- suppression du client fictif
DELETE FROM CLIENT WHERE email = 'missuji@gmail.com' ;
```



Chaque test a été exécuté **individuellement**, la levée d'une exception interrompant l'exécution du programme. Il n'est de ce fait pas possible de tester plusieurs appels à la procédure au cours de la même exécution. Pour exécuter le programme de test présenté, il est donc nécessaire de commenter l'ensemble des appels à la procédure non considérés et de laisser "décommenté" l'appel à tester.



La variable `@varIdClientTest1` contient le numéro (ID) d'un client enregistré dans la base de données.



## RÉSULTAT DES TESTS

N° TEST	ERREUR ATTENDUE	RÉSULTAT
<i>CLIENT</i>		
[10]	#45010 - Erreur : Client non renseigné.	OK
[11]	#45011 - Erreur : Client inconnu.	OK
<i>MODE DE LIVRAISON</i>		
[20]	#45020 - Erreur : Mode de livraison non renseigné.	OK
[21]	#45021 - Erreur : Mode de livraison invalide.	OK
<i>ADRESSE DE LIVRAISON</i>		
[30]	#45030 - Erreur : Adresse de livraison non renseignée.	OK
[31]	#45031 - Erreur : Adresse ne correspondant à aucun domicile enregistré.	OK
[32]	#45032 - Erreur : Adresse ne correspondant à aucun point relais enregistré.	OK
<i>MODE DE PAIEMENT</i>		
[40]	#45040 - Erreur : Mode de paiement non renseigné.	OK
[41]	#45041 - Erreur : Mode de paiement invalide.	OK
<i>CARTE BANCAIRE</i>		
[50]	#45050 - Erreur : Carte bancaire non renseignée.	OK
[51]	#45051 - Erreur : Carte bancaire inconnue.	OK
<i>NUMÉRO PAYPAL</i>		
[60]	#45060 - Erreur : Numéro PayPal non renseigné.	OK
<i>NUMÉRO DE VIREMENT</i>		
[70]	#45070 - Erreur : Numéro de virement non renseigné.	OK

## 2.2 Procédures de gestion des regroupements

### 2.2.1 Procédure d'ajout d'un regroupement

#### Spécifications techniques

La procédure SQL/PSM **CreerRegroupement** répond aux spécifications suivantes :

#### Entrées :

- **p\_listProduit** : Une liste des **idProduit**.
- **p\_idR** : **idRegroupement** auquel les produits doivent être associés.

#### Actions :

- Vérification de l'existence du regroupement et des produits.
- Traitement des produits : la liste des produits est parcourue, ID par ID, pour détecter les séparateurs (virgules).
- Ajout des associations dans la table **REGROUPER** : Si le regroupement et le produit existe, une ligne est insérée dans la table **REGROUPER** pour lier le produit au regroupement.
- Validation de la transaction (**COMMIT**).

#### Gestion d'erreurs :

- Regroupement inconnu : Si le regroupement n'existe pas, un message d'erreur est levé.
- Produit inconnu : Si un produit de la liste n'existe pas, un message d'erreur est levé.

#### Code de la procédure

##### PROCÉDURE SQL/PSM

```
DELIMITER $$

CREATE PROCEDURE CreerRegroupement(IN p_listProduit VARCHAR(255), IN p_idR INT)
BEGIN
    DECLARE v_idVariete INT ;
    DECLARE v_pos INT DEFAULT 1 ;
    DECLARE v_nextPos INT ;
    DECLARE v_variete INT ;
    DECLARE v_count INT ;
    SELECT COUNT(*) INTO v_count
    FROM REGROUPEMENT
    WHERE idRegroupement = p_idR ;
    IF v_count = 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT= 'Erreur : regroupement inconnu !' ;
    END IF ;
    WHILE v_pos <= LENGTH(p_listProduit) DO
        SET v_nextPos = LOCATE(',', p_listProduit, v_pos) ;
```

```

IF v_nextPos = 0 THEN
    SET v_variete = SUBSTRING(p_listProduit, v_pos) ;
    SET v_pos = LENGTH(p_listProduit) + 1 ;
ELSE
    SET v_variete = SUBSTRING(p_listProduit, v_pos, v_nextPos - v_pos) ;
    SET v_pos = v_nextPos + 1 ;
END IF ;
SET v_idVariete = CAST(v_variete AS UNSIGNED) ;
SELECT COUNT(*) INTO v_count
FROM VARIETE
WHERE idVariete = v_idVariete ;
IF v_count = 0 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT= 'Erreur : produit inconnu !' ;
END IF ;
INSERT INTO REGROUPER (idRegroupement, idVariete) VALUES (p_idR, v_idVariete);
END WHILE ;
COMMIT ;
END$$

DELIMITER ;

```

## Test de la procédure

### Cas nominal

#### AFFICHAGE DU CONTENU DE LA TABLE REGROUPEMENT

+ Options

					idRegroupement	nomRegroupement
<input type="checkbox"/>		Éditer		Copier		Supprimer
					1	Nouveautés
<input type="checkbox"/>		Éditer		Copier		Supprimer
					2	Soldes
<input type="checkbox"/>		Éditer		Copier		Supprimer
					3	Made in France
<input type="checkbox"/>		Éditer		Copier		Supprimer
					4	Meilleures Ventes
<input type="checkbox"/>		Éditer		Copier		Supprimer
					5	Offres Spéciales
<input type="checkbox"/>		Éditer		Copier		Supprimer
					6	Édition Limitée

#### EXÉCUTION DE LA PROCÉDURE

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0259 seconde(s).)

CALL CreerRegroupement( '45,32,89,67,13', 4);

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## AFFICHAGE DU CONTENU DE LA TABLE REGROUPER

+ Options

					<span style="color: blue;">idRegroupement</span>	<span style="color: blue;">idProduit</span>
<input type="checkbox"/>	Éditer	Copier	Supprimer	4		13
<input type="checkbox"/>	Éditer	Copier	Supprimer	4		32
<input type="checkbox"/>	Éditer	Copier	Supprimer	4		45
<input type="checkbox"/>	Éditer	Copier	Supprimer	4		67
<input type="checkbox"/>	Éditer	Copier	Supprimer	4		89

### Cas d'erreur

NUMÉRO DE PRODUIT (idVariete) INCONNU

#### Erreur

Requête SQL : [Copier](#)

```
CALL CreerRegroupement('234, 500, 1234', 1);
```

MySQL a répondu :

#1644 - Erreur : produit inconnu !

NUMÉRO DE REGROUPEMENT (idRegroupement) INCONNU

#### Erreur

Requête SQL : [Copier](#)

```
CALL CreerRegroupement('23, 67, 9', 55);
```

MySQL a répondu :

#1644 - Erreur : regroupement inconnu !

## 2.2.2 Procédure de suppression d'un regroupement

### Spécifications techniques

La procédure SQL/PSM `SupprimerRegroupement` répond aux spécifications suivantes :

#### Entrées :

- `p_idR` : idRegroupement à supprimer.

#### Actions :

- Vérification de l'existence du regroupement.
- Suppression des relations : toutes les associations liées à ce regroupement dans la table `REGROUPER` sont supprimés.
- Validation de la transaction (`COMMIT`).

#### Gestion d'erreurs :

- Regroupement inconnu : Si idRegroupement ne correspond à aucun regroupement dans la table `REGROUPEMENT`, une erreur est levée.

### Code de la procédure

#### PROCÉDURE SQL/PSM

```
DELIMITER $$

CREATE PROCEDURE SupprimerRegroupement(IN p_idR INT)
BEGIN
    DECLARE v_count INT ;

    SELECT COUNT(*) INTO v_count
    FROM REGROUPEMENT
    WHERE idRegroupement = p_idR ;

    IF v_count = 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT= 'Erreur : regroupement inconnu !' ;
    END IF ;
    DELETE FROM REGROUPER WHERE idRegroupement = p_idR ;
    COMMIT ;
END$$

DELIMITER ;
```

## Test de la procédure

### Cas nominal

#### EXÉCUTION DE LA PROCÉDURE

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0104 seconde(s).)

```
CALL SupprimerRegroupement(4);
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

#### AFFICHAGE DU CONTENU DE LA TABLE REGROUPER

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0029 seconde(s).)

```
SELECT * FROM REGROUPER WHERE idRegroupement = 4;
```

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

idRegroupement	idProduit
----------------	-----------

Opérations sur les résultats de la requête

### Cas d'erreur

#### NUMÉRO DE REGROUPEMENT (idRegroupement) INCONNU

##### Erreur

Requête SQL : [Copier](#)

```
CALL SupprimerRegroupement(45);
```

MySQL a répondu : ⓘ

#1644 - Erreur : regroupement inconnu !



**Équipe 2B12**

Nolhan Biblocque

Léo Guinvarc'h

Victor Jockin

Mathys Laguilliez

Mucahit Lekesiz

**À destination de**

Laurent Nonne

**Formation**

BUT Informatique

2ème Année

Promotion 2024-2025

**Établissement**

IUT de Blagnac

Université Toulouse II – Jean Jaurès (31)