



# Ludorama

Votre monde de jeu, sans frontière

Pierre CORNU | Raphaël LAMOTHE | Naria SAVARY | David TRAN  
Groupe 1A

# TABLE DES MATIÈRES

---

<b>1 - Contexte.....</b>	<b>3</b>
<b>2-Procédures.....</b>	<b>4</b>
Généralité.....	4
get_commande_details.....	5
get_dispos_produits.....	5
get_dispos_produits_light.....	6
get_client_cb.....	6
get_client_paypal.....	7
del_cb.....	7
del_paypal.....	8
get_moyenne_prod.....	8
get_panier.....	9
<b>3-Triggers.....</b>	<b>10</b>



## 1 – CONTEXTE

Ce document fait suite au premier compte rendu concernant la BD, il le complète en ajoutant à ce dernier les informations sur les procédures stockées qui y ont été intégrées. De plus, considérant l'impossibilité d'ajouter des triggers à la base de données à cause de problèmes de permission, ce document traitera de pistes sur les triggers qui auraient dû être implémentées mais ne fournira pas le code qui va avec car le temps qui aurait dû y être attribué à été alloué ailleurs.

Note : Puisque le rendu de ce document est dû avant la fin du projet, il n'est pas exclu que certaines procédures soient ajoutées à posteriori. Il en va de même pour le document précédent : la base de données n'a pas fondamentalement changé depuis, mais elle a subi quelques évolutions. Dans le cadre d'un projet réel, ces documents auraient été tenu à jour et rendu après la fin du développement, évitant ces légers écarts entre le document rendu et la réalité.



## 2-PROCÉDURES

La philosophie que nous avons adoptée dans le développement des procédures est d'épurer le code PHP en enlevant les appels à des requêtes SQL trop chargées. Ainsi, nous avons remplacé par des procédures toutes les requêtes qui ne sont pas "simples", c'est-à-dire toutes les requêtes qui ne sont pas sur une seule table. Certaines procédures ont été faites en prévision (comme `get_commande_details`) afin d'être disponibles dès que le besoin est rencontré. Certaines procédures moins évidentes car dépendantes des solutions techniques utilisées par les membres du groupe (comme `get_client_cb`) ont été programmées au moment où le besoin a été rencontré, tout en maintenant le script à jour. C'est de par cette stratégie qu'il existe un risque que ce document ne soit plus à jour à la fin du projet.

### Généralité

Pour épurer le document, une partie du script qui est commune à toutes les procédures sera décrite ici et ne sera plus rappelée dans le détail de chaque procédure. Ainsi, toutes les procédures dans le script sont enrobées ainsi :

```
DROP PROCEDURE <nom_procédure>;
```

```
DELIMITER $$
```

```
<code_procédure>;
```

```
DELIMITER ;
```



## get\_commande\_details

Cette procédure, donnée l'id d'une commande permet d'obtenir la liste des produits qui sont concernés par la commande avec le nombre d'articles à chaque fois.

```
CREATE PROCEDURE get_commande_details(IN commande_id INT)
BEGIN
    SELECT IDPROD, NOMPROD, QTE
    FROM PRODUIT P, CONTIENT C
    WHERE P.IDPROD = C.IDPROD
    AND C.IDCOMMANDE = commande_id;
END$$
```

## get\_dispos\_produits

Cette procédure, donnée l'id d'un produit, fournit toutes les combinaisons disponibles pour ce produit, c'est-à-dire les couleurs, les conditionnements et les formats dans lequel le produit est disponible ainsi que le prix qui va avec.

```
CREATE PROCEDURE get_dispos_produit(IN produit_id INT)
BEGIN
    SELECT DF.IDPROD, NOMFORMAT, PRIX, NOMCOULEUR, NOMCONDI
    FROM DISPOFORMAT DF, FORMATPROD F, DISPONIBLECONDITIONNEMENT DC,
    CONDITIONNEMENT C, DISPONIBLECOULEUR DC2, COULEUR C2
    WHERE F.IDFORMAT = DF.IDFORMAT AND DF.IDPROD = produit_id
    AND C.IDCONDI = DC.IDCONDI AND DC.IDPROD = produit_id
    AND C2.IDCOULEUR = DC2.IDCOULEUR AND DC2.IDPROD = produit_id;
END$$
```

## get\_dispos\_produits\_light

Cette procédure, donnée l'id d'un produit, fonctionne comme la précédente mais en ne fournissant que les formats disponibles avec le prix associé.

```
CREATE PROCEDURE get_dispos_produit_light(IN produit_id INT)
BEGIN
    SELECT DF.IDPROD, NOMFORMAT, PRIX
    FROM DISPOFORMAT DF, FORMATPROD F
    WHERE F.IDFORMAT = DF.IDFORMAT AND DF.IDPROD = produit_id;
END$$
```

## get\_dispos\_produit\_borne

Cette procédure, donnée une borne inférieure et une borne supérieure, fournit tous les articles et leur format qui sont compris entre les deux bornes.

```
CREATE PROCEDURE get_dispos_produit_borne(IN borne_inf FLOAT(6,2), IN
borne_sup FLOAT(6,2))
BEGIN
    SELECT DF.IDPROD, NOMFORMAT, PRIX
    FROM DISPOFORMAT DF, FORMATPROD F
    WHERE PRIX >= borne_inf AND PRIX <= borne_sup;
END$$
```

## get\_client\_cb

Cette procédure, donnée l'id d'un client fournit les informations concernant toutes les cartes bleues associées au client.

```
CREATE PROCEDURE get_client_cb(IN client_id INT)
BEGIN
    SELECT CB.*
    FROM CB, METHODEPAIEMENT M
    WHERE CB.NUMCARTE = M.NUMCARTE
    AND M.IDCOMPTE = client_id;
END$$
```

## get\_client\_paypal

Cette procédure, donnée l'id d'un client fournit les informations concernant tous les comptes paypal associés au client.

```
CREATE PROCEDURE get_client_paypal(IN client_id INT)
BEGIN
    SELECT PAYPAL.*
    FROM PAYPAL, METHODEPAIEMENT M
    WHERE PAYPAL.IDPAYPAL= M.IDPAYPAL
    AND M.IDCOMPTE = client_id;
END$$
```

## del\_cb

Cette procédure, donnée l'id d'une carte bleue, supprime cette dernière ainsi que la MéthodePaiement qui la représente

```
CREATE PROCEDURE del_cb(IN id_cb CHAR(16))
BEGIN
    DELETE FROM METHODEPAIEMENT
    WHERE NUMCARTE = id_cb;
    DELETE FROM CB
    WHERE NUMCARTE = id_cb;
END$$
```

## del\_paypal

Cette procédure, donnée l'id d'un compte paypal, supprime ce dernier ainsi que la MéthodePaiement qui le représente.

```
CREATE PROCEDURE del_paypal(IN id_paypal INT)
BEGIN
    DELETE FROM METHODEPAIEMENT
    WHERE IDPAYPAL = id_paypal;
    DELETE FROM PAYPAL
    WHERE IDPAYPAL = id_paypal;
END$$
```





## get\_moyenne\_prod

Cette procédure, donnée l'id d'un produit, fournit la moyenne des notes d'utilisateur qu'il a reçues. Cela doit fournir un réel compris entre 0 et 10 qui représente le "nombre de demi étoiles"

```
CREATE PROCEDURE get_moyenne_prod(IN produit_id INT)
BEGIN
    SELECT AVG(SUM(NBETOILE)) AS noteAvg
    FROM COMMENTAIRE
    WHERE IDPROD = produit_id
    GROUP BY IDPROD;
END$$
```

## get\_panier

Cette procédure, donnée l'id d'un client, retourne l'id de la commande qui a le statut "Panier" associé à ce compte.

```
CREATE PROCEDURE get_panier(IN id_compte INT)
BEGIN
    SELECT IDCOMMANDE
    FROM COMMANDE
    WHERE IDCOMPTE = id_compte
    AND STATUSCOMMANDE = 'Panier';
END$$
```



## Tests :

-- Tests de la procédure get\_commande\_details

CALL get\_commande\_details(1);

CALL get\_commande\_details(2);

-- Tests de la procédure get\_dispos\_produit

CALL get\_dispos\_produit(10);

CALL get\_dispos\_produit(15);

-- Tests de la procédure get\_dispos\_produit\_light

CALL get\_dispos\_produit\_light(10);

CALL get\_dispos\_produit\_light(15);

-- Tests de la procédure get\_dispos\_produit\_borne

CALL get\_dispos\_produit\_borne(10.00, 20.00);

CALL get\_dispos\_produit\_borne(5.00, 15.00);

-- Tests de la procédure get\_client\_cb

CALL get\_client\_cb(1);

CALL get\_client\_cb(2);

-- Tests de la procédure get\_client\_paypal

CALL get\_client\_paypal(1);

CALL get\_client\_paypal(2);

-- Tests de la procédure del\_cb

CALL del\_cb('1010101010101010');

CALL del\_cb('9999888877776666');

-- Tests de la procédure del\_paypal

CALL del\_paypal(1);

CALL del\_paypal(2);

-- Tests de la procédure get\_moyenne\_prod

CALL get\_moyenne\_prod(10);

CALL get\_moyenne\_prod(15);

-- Tests de la procédure get\_panier

CALL get\_panier(1);

CALL get\_panier(2);

## 3-TRIGGERS

Comme mentionné précédemment, dans le contexte où les permissions ne permettent pas d'utiliser des triggers dans la base, nous n'avons pas développé ces derniers. Cependant il y en avait plusieurs qui nous semblaient importants, certains même dès la conception initiale de la BD. Voici donc les triggers qui auraient été importants à notre sens.

- Un trigger pour contrôler qu'une MethodePaiement est associée à une option de paiement concrète et non pas deux et qui vérifie que l'OptionPaiement associée est mise à jour correctement lorsqu'il y a des changements.
- Un trigger permettant de supprimer les associations produits commande dont la quantité est  $\leq 0$
- Il est envisageable d'ajouter un trigger qui vérifie qu'un utilisateur a bien commandé un article pour ajouter un commentaire