

Présentation

Le sujet

Une société souhaite une refonte de son système d'information (SI) et l'automatisation d'une partie du SI, avec principalement la création d'un site (ou application) de présentation et de ventes de produits en ligne. Le site devra être hébergé sur des serveurs. La société sera responsable des serveurs et du déploiement de l'application sur ces derniers. L'ensemble devra répondre aux contraintes de sécurité, de performance et aux aspects juridiques sur la protection des données.

Numéro de l'équipe

eq_02_01

Membres de l'équipes

OULMAS Thomas : Étude sur la structure de la Base de Données

ROUX Célestin : Rédaction des procédures

VAN LAERE Pierric : Étude sur la structure de la Base de Données

MOREAU-THOMAS Nils : Rédaction des procédures

Le client

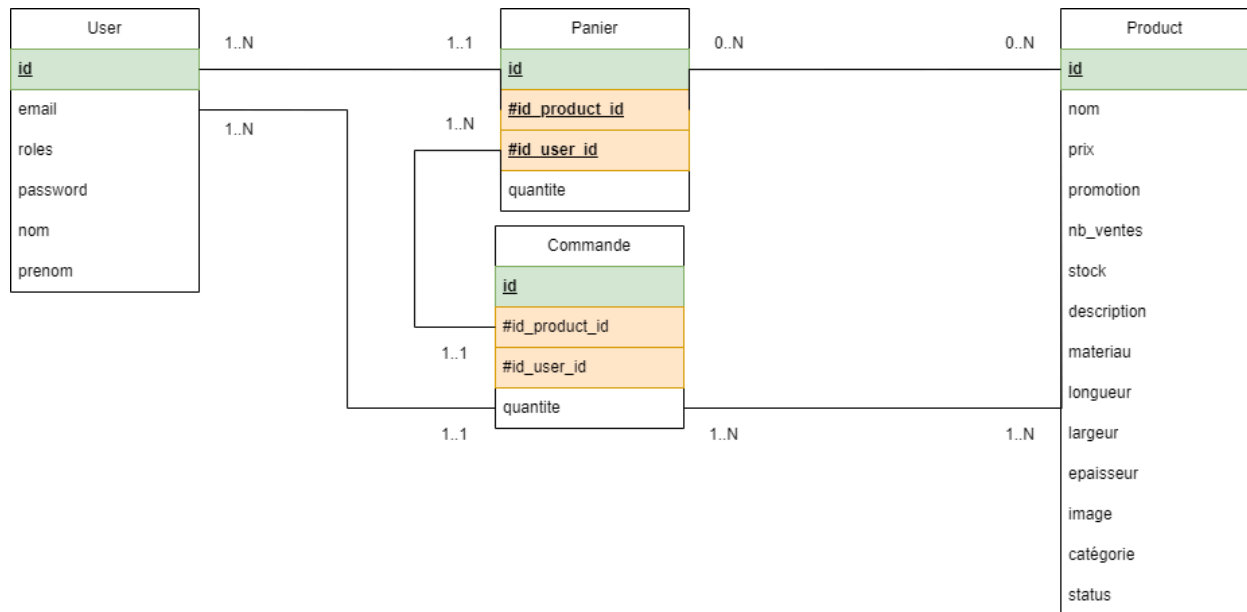
Le client de notre site se nomme Antoine, il possède une graveuse laser avec laquelle il grave des pièces sur mesures en bois. Mais il vend aussi des pièces uniques qu'il produit en parallèle des demandes de ses clients.

Ces pièces sont des dessous de verres, des boules de Noël, des portes-clés et autres.

Il possède :

- un instagram : @antoinefactory
- une adresse mail : antoinefactory@gmail.com

Les tables et leurs relations



Disponible ici :

https://drive.google.com/drive/folders/1yne7hV8fWDcLLMmfrpJ2FY6_dixnO-3I?usp=sharing

Les procédures stockées

Note

Notre site utilise mariaDB.

Il est important de noter que ces procédures ont été écrites à l'aide d'une documentation en ligne car la syntaxe diffère d'Oracle et certaines choses (comme le "L00P") n'ont pas été vu en cours.

Ces procédures ci-dessous n'ont pas été utilisées dans le site, puisque le framework Symfony les effectue déjà par le biais d'entités, qui une fois modifiées, modifient les ligne associées dans la base de données, et la logique de certaines ont déjà été implémentées en php.

Le fichier dans son entièreté est disponible ici :

https://drive.google.com/file/d/1C1WhqQkPJFTqTHmUTpnq50Rd3xDMfps-/view?usp=drive_link

Suppression d'un utilisateur

```
/*procédure n1*/  
CREATE OR REPLACE PROCEDURE supprimer_user(IN user_id INT)
```

```

BEGIN
    DECLARE cnt INT;
    SELECT COUNT(*) INTO cnt FROM commande WHERE id_user_id = user_id;

    IF cnt > 0 THEN
        SIGNAL SQLSTATE '45001'
        SET MESSAGE_TEXT = 'L utilisateur possede des commandes en cours';
    ELSE
        DELETE FROM panier WHERE id_user_id = user_id;
        DELETE FROM commande WHERE id_user_id = user_id;
        DELETE FROM user WHERE id = user_id;
        SELECT 'L utilisateur est supprimé';
    END IF;
END;

/*insert test1*/
INSERT INTO user VALUES(999,'user@test.fr
','["ROLE_USER"]','$2y$13$Km3pXs/NW6NFMr9JrdqZH0Kij1SQJyhRctUECm2RA0EgZJa1
AUPYW','Letest','Utilisateur');

/*test1*/
PREPARE stmt1 FROM 'CALL supprimer_user(?)';
SET @a = 999;
EXECUTE stmt1 USING @a;

```

Cette procédure fonctionne comme ceci :

Quand un utilisateur est supprimer

- si jamais il est dans la table commande, alors la suppression ne se fait pas
- sinon toutes les lignes contenant son id sont supprimer

Suppression d'un articles

```

/*procédure n2*/
CREATE OR REPLACE PROCEDURE supprimer_article(IN product_id INT)

BEGIN
    DECLARE cnt INT;
    SELECT COUNT(*) INTO cnt FROM commande WHERE id_product_id =
product_id;

    IF cnt > 0 THEN
        SIGNAL SQLSTATE '45002'
        SET MESSAGE_TEXT = 'L article est contenu dans des commandes en

```

```

cours';
    ELSE
        DELETE FROM panier WHERE id_product_id = product_id;
        DELETE FROM commande WHERE id_product_id = product_id;
        DELETE FROM product WHERE id = product_id;
        SELECT 'L article est supprimé';
    END IF;
END;

/*insert test2*/
INSERT INTO product VALUES(999,'Test',54.0,2,72,1,'Un
test','Bois',84.0,36.0,77.0,'image.jpg','Porte-clé','low');

/*test2*/
PREPARE stmt2 FROM 'CALL supprimer_article(?)';
SET @a = 999;
EXECUTE stmt2 USING @a;

```

Cette procédure fonctionne comme ceci :

Quand un article est supprimer

- si jamais il est dans la table commande, alors la suppression ne se fait pas
- sinon toutes les lignes contenant son id sont supprimer

Ajout d'un article dans panier

```

CREATE OR REPLACE PROCEDURE ajout_panier(IN p_product_id INT, IN p_user_id
INT, IN p_quantite INT)
BEGIN
    DECLARE pan_item_count INT;

    /*regarde si l article est déjà dans le panier de user*/
    SELECT COUNT(*) INTO pan_item_count FROM panier WHERE id_product_id
= p_product_id AND id_user_id = p_user_id;

    IF pan_item_count > 0 THEN
        /*l article est déjà dans le panier de user*/
        UPDATE panier
        SET quantite = quantite + p_quantite
        WHERE id_product_id = p_product_id AND id_user_id = p_user_id;
    ELSE
        /*l article n'est pas déjà dans le panier de user*/

```

```

        INSERT INTO panier (id_product_id, id_user_id, quantite)
        VALUES (p_product_id, p_user_id, p_quantite );
    END IF;
END;

/*insert test3*/
INSERT INTO product VALUES(999,'Test',54.0,2,72,1,'Un
test','Bois',84.0,36.0,77.0,'image.jpg','Porte-clé','low');

INSERT INTO user VALUES(999,'user@test.fr
','["ROLE_USER"]','$2y$13$Km3pXs/NW6NFMr9JrdqZH0Kij1SQJyhRctUECm2RA0EgZJa1
AUPYW','Letest','Utilisateur');

/*test3*/
PREPARE stmt3 FROM 'CALL ajout_panier(?,?,?)';
SET @a = 999;
SET @b = 999;
SET @c = 5;
EXECUTE stmt3 USING @a,@b,@c;

```

Cette procédure fonctionne comme ceci :

Quand un utilisateur ajoute au panier un article

- si jamais l'article est déjà dans son panier, alors la quantité dans le panier sera celle déjà dans le panier + celle que l'on ajoute
- sinon une ligne se crée avec l'article et l'utilisateur qui l'a commandé

Passage de la commande d'un utilisateur

```

CREATE PROCEDURE passage_commande(IN p_user_id INT)
BEGIN
    DECLARE product_id, pan_quantite, available_stock INT;
    DECLARE @done BOOLEAN DEFAULT FALSE; -- Add this line

    DECLARE pan_cursor CURSOR FOR
    SELECT p.id_product_id, p.quantite, pr.stock FROM panier p JOIN product pr ON p.id_product_id = pr.id WHERE p.id_user_id = p_user_id;

    DECLARE CONTINUE HANDLER FOR NOT FOUND --si le curseur arrive à la fin
    SET @done = TRUE;

    OPEN pan_cursor;

    pan_loop: LOOP
    FETCH pan_cursor INTO product_id, pan_quantite, available_stock;

```

```

IF @done THEN --le curseur à fini
    LEAVE pan_loop;
END IF;

IF pan_quantite > available_stock THEN
    SIGNAL SQLSTATE '45003'
    SET MESSAGE_TEXT = 'La quantité commandée n est pas en stock';
ELSE
    -- Ajout dans la table commande
    INSERT INTO commande (id_product_id, id_user_id, quantite)
    VALUES (product_id, p_user_id, pan_quantite);

    -- Update product pour montrer la vente
    UPDATE product
    SET stock = stock - pan_quantite,
        nb_ventes = nb_ventes + pan_quantite
    WHERE id = product_id;
END IF;
END LOOP;

CLOSE pan_cursor;
END;

/*insert test4*/
INSERT INTO product VALUES(999,'Test',54.0,2,72,1,'Un test','Bois',84.0,36.0,77.0,'image.jpg','Porte-clé','low');

INSERT INTO user VALUES(999,'user@test.fr',
    '','["ROLE_USER"]','$2y$13$Km3pXs/NW6NFMr9JrdqZH0Kij1SQJyhRctUECm2RA0EgZJa1AUPYW','Letest','Utilisateur');

PREPARE stmt4 FROM 'CALL ajout_panier(?,?,?)';
SET @a = 999;
SET @b = 999;
SET @c = 5;
EXECUTE stmt4 USING @a,@b,@c;

/*test4*/
PREPARE stmt5 FROM 'CALL passage_commande(?)';
SET @d = 999;
EXECUTE stmt5 USING @d;

```

Cette procédure fonctionne comme ceci :

Quand un utilisateur passe une commande, on regarde chaque article de son panier

- si jamais la quantité dans le panier excède celle disponible alors l'article n'est pas commandé
- sinon il est commandé et le nombre de vente augmente + le stock disponible diminue