

Livrable semaine 51

Appel des procédures stockées



Introduction.....	3
Chargement des produits parents.....	4
1) Code.....	4
Trace d'exécution côté php.....	4
2) Trace côté site (visuelle).....	5
Chargement des photos de produits.....	7
1) Code.....	7
2) Trace d'exécution côté php.....	8
3) Trace côté site (visuelle).....	9
Chargement du top 4 des produits à l'accueil.....	11
1) Code.....	11
2) Trace d'exécution côté php.....	11
3) Trace côté site (visuelle).....	13
Chargement des catégories enfants d'une catégorie.....	15
1) Code.....	15
2) Trace d'exécution côté php.....	16
3) Trace côté site (visuelle).....	18
Chargement de tous les produits d'une catégorie.....	21
1) Code.....	21
2) Trace d'exécution côté php.....	22
3) Trace côté site (visuelle).....	24
Chargement des produits apparentés à un produit.....	26
1) Code.....	26
2) Trace d'exécution côté php.....	27
3) Trace côté site (visuelle).....	29
Savoir si un client est autorisé à déposer un avis sur un produit.....	32
1) Code.....	32
2) Trace d'exécution côté php.....	33
3) Trace côté site (visuelle).....	34
Chargement de toutes les infos d'un produit décliné.....	38
1) Code.....	38
2) Trace d'exécution côté php.....	38
3) Trace côté site (visuelle).....	40
Conclusion.....	42

Introduction

L'utilisation systématique de procédures stockées permet de **garantir la cohérence des données** et de centraliser la **logique métier côté base de données**.

Les exceptions SQL sont interceptées côté PHP afin d'assurer la continuité de service, avec **des mécanismes de repli (tableaux vides, placeholders, redirections)** garantissant un **affichage maîtrisé et sécurisé**, même en cas de paramètres invalides ou de données absentes.

Précision sur la levée d'exceptions :

Les procédures stockées SQL disposant de **paramètres d'entrée** ont été **volontairement programmées pour lever des exceptions SQL** lorsque l'un ou plusieurs **paramètres fournis sont invalides** (par exemple : identifiant de client inexistant, identifiant de produit inexistant, incohérence métier).

Ces exceptions sont **interceptées à l'aide de blocs try/catch** côté PHP, **permettant une gestion contrôlée et sécurisée des erreurs**.

Chargement des produits parents

1) Code

```
function chargerProduitsParents()  
{  
    $tab_produits = [];  
    try {  
        global $connection;  
        $statement = $connection->prepare("CALL getProduitsParents()");  
        //Récupération de tous les produits qui ne sont PAS des KITS, et on  
        récup les catégories aussi  
        $statement->execute(); //Executer la commande  
        while ($ligne = $statement->fetch((PDO::FETCH_ASSOC))) {  
            //Récup un tableau associatif  
            $tab_produits[$ligne['idProduit']] = $ligne; //Tableau  
            associant l'id du produit a toutes ses infos  
        }  
        $statement->closeCursor();  
    } catch (PDOException $e) {  
        error_log("Erreur SQL chargerProduitsParents : " .  
$e->getMessage());  
    } return $tab_produits; } //On renvoie le tableau des des produits
```

Trace d'exécution côté php

Code de test de la procédure :

```
//Appel de la fonction  
$produitsParents = chargerProduitsParents();  
//Affichage du résultat  
echo "<pre>";  
var_dump($produitsParents);  
echo "</pre>";
```

Résultat obtenu sans exception :

```
array(51) {
  [1]=>
  array(4) {
    ["idProduit"]=>
    int(1)
    ["libelle"]=>
    string(14) "Potion de mana"
    ["description"]=>
    string(74) "Fiole contenant un liquide bleu lumineux permettant de restaurer la magie."
    ["categorie"]=>
    string(4) "Mana"
  }
  [2]=>
  array(4) {
    ["idProduit"]=>
    int(2)
    ["libelle"]=>
    string(21) "Potion de résistance"
    ["description"]=>
    string(76) "Potions alchimiques offrant une protection temporaire contre les éléments."
    ["categorie"]=>
    string(27) "Résistances Élémentaires"
  }
  [3]=>
  array(4) {
    ["idProduit"]=>
    int(3)
    ["libelle"]=>
    string(14) "Potion de soin"
    ["description"]=>
    string(68) "Le meilleur ami de l'aventurier, referme les plaies instantanément."
    ["categorie"]=>
    string(5) "Soins"
  }
  [4]=>
```

Résultat obtenu si une exception survient (un tableau vide est renvoyé) :

```
array(0) {
}
```

2) Trace côté site (visuelle)

Extrait du code gérant l'erreur :

```
<!--Si au moins 1 produit est trouvé-->
<?php if(count($tab_produits) > 0): ?>
<!--Génération du tableau des produits.....-->
<!--. . . . .-->
    <!--. . . . .-->
<?php else :
    echo "Aucun produit disponible !";
endif; ?>
```

Résultat obtenu sans exception :

Le Grimoire (Produits)

+ Créer un Produit (Parent)

ID	NOM DU PRODUIT (ABSTRAIT)	CATÉGORIE	VARIANTES
#1	> Potion de mana	MANA	2 déclinaisons
#2	> Potion de résistance	RÉSISTANCES ÉLÉMENTAIRES	4 déclinaisons
#3	> Potion de soin	SOINS	2 déclinaisons
#4	> Manuel d'informations	GUIDES PRATIQUES	2 déclinaisons

Résultat obtenu si une exception survient :

Le Grimoire (Produits)

+ Créer un Produit (Parent)

Aucun produit disponible !

Chargement des photos de produits

1) Code

```
/**
 * Fonction permettant de charger les photos d'un produit
 */
function chargerPhotosProduit($idDeclinaison, $libelleDeclinaison)
{
    global $connection;
    $photos = [];

    try {
        $stmt = $connection->prepare("CALL
getPhotosProduit(:idDeclinaison)");
        $stmt->execute(['idDeclinaison' => $idDeclinaison]);
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $photos[] = $row['url'];
        }
        $stmt->closeCursor();
    } catch (PDOException $e) { //La procédure déclenche une exception
si aucune photo trouvée. On s'en sert de fallback pour générer un
placeholder
        $photos[] = 'https://placeholder.co/400x400/228B22/FFFFFF?text='
. $libelleDeclinaison; //Afficher un placeholder si l'image n'est pas
trouvée / une quelconque exception est survenue
    }

    return $photos;
}
```

Précision : Si la procédure stockée `getPhotosProduit(:idDeclinaison)` est appelée avec un id de déclinaison ne possédant pas de photo, alors cette dernière lève une exception SQL, permettant la génération automatique de l'image “placeholder” de secours.

2) Trace d'exécution côté php

Cas 1 : ID de la déclinaison existant, et des photos sont trouvées

```
//Appel de la fonction
$idDeclinaison = 2;
$libelleDeclinaison = "Potion Mana Mineure";
$photosProduit = chargerPhotosProduit($idDeclinaison,$libelleDeclinaison);
echo "<pre>";
var_dump($photosProduit);
echo "</pre>";
```

Obtenu :

```
array(4) {
  [0]=>
  string(23) "images/manaMineure1.jpg"
  [1]=>
  string(23) "images/manaMineure2.jpg"
  [2]=>
  string(23) "images/manaMineure3.jpg"
  [3]=>
  string(23) "images/manaMineure4.jpg"
}
```

Cas 2 : ID de la déclinaison existant, et des photos ne sont pas trouvées : une exception est levée et une image de fallback est générée

```
//Appel de la fonction
$idDeclinaison = 1;
$libelleDeclinaison = "Potion Mana Majeure";

$photosProduit =
chargerPhotosProduit($idDeclinaison,$libelleDeclinaison);

echo "<pre>";
var_dump($photosProduit);
echo "</pre>";
```

Obtenu :

```
array(1) {
  [0]=>
  string(67) "https://placeholder.co/400x400/228B22/FFFFFF?text=Potion Mana Majeure"
}
```


Cas 3 : ID de la déclinaison inexistant, la procédure lève une exception attrapée, et une image avec le texte du libellé est généré

```
//Appel de la fonction
$idDeclinaison = 999;
$libelleDeclinaison = "Produit";

$photosProduit =
chargerPhotosProduit($idDeclinaison,$libelleDeclinaison);

echo "<pre>";
var_dump($photosProduit);
echo "</pre>";
```

Obtenu :

```
array(1) {
  [0]=>
  string(55) "https://placeholder.co/400x400/228B22/FFFFFF?text=Produit"
}
```

3) Trace côté site (visuelle)

Code php / html appelant la procédure :

```
<div class="w-full <?php echo $hauteurImage; ?> bg-gradient-to-br
from-purple-900 to-purple-700 rounded-lg flex items-center justify-center mb-3
glow-green overflow-hidden">
    " class="w-full h-full
object-cover rounded-lg"
onerror="this.onerror=null;this.src='https://placeholder.co/400x400/228B22/FFFFF
F?text=Image introuvable';" />
</div>
```

Cas 1 : ID de la déclinaison existant, et des photos sont trouvées

Obtenu :

L'image est bien disponible pour ce produit, elle a été chargée correctement.



Cas 2 : ID de la déclinaison existant, et des photos ne sont pas trouvées : une exception est levée et une image de fallback est générée

Obtenu :

Aucune image n'est disponible pour ce produit, le placeholder contenant le libellé du produit à bien été généré.



Chargement du top 4 des produits à l'accueil

1) Code

```
function getTopProduitsAccueil()  
{  
    $vedettes = [];  
    global $connection;  
    // --- Appel de la procédure stockée pour les produits vedettes ---  
    try {  
        $stmt = $connection->query("CALL getTopProduitsAccueil()");  
        $vedettes = $stmt->fetchAll(PDO::FETCH_ASSOC);  
  
        // Important pour éviter les erreurs avec PDO + procédures  
stockées  
        while ($stmt->nextRowset()) {  
            // vider les résultats restants  
        }  
    } catch (PDOException $e) {  
        error_log("Erreur SQL getTopProduitsAccueil : " .  
$e->getMessage());  
    }  
  
    return $vedettes;  
}
```

2) Trace d'exécution côté php

Code de test de la procédure :

```
//Appel de la fonction  
$topProduits = getTopProduitsAccueil();  
echo "<pre>";  
var_dump($topProduits);  
echo "</pre>";
```

Résultat obtenu sans exception :

```
array(4) {
  [0]=>
  array(8) {
    ["idProduit"]=>
    int(2)
    ["nomProduit"]=>
    string(21) "Potion de résistance"
    ["idDeclinaison"]=>
    int(5)
    ["prix"]=>
    string(4) "9.99"
    ["prixSolde"]=>
    NULL
    ["qualite"]=>
    string(10) "Peu commun"
    ["libelleDeclinaison"]=>
    string(28) "Potion de Résistance au feu"
    ["descDeclinaison"]=>
    string(33) "Résistance aux chaleurs intenses"
  }
  [1]=>
  array(8) {
    ["idProduit"]=>
    int(5)
    ["nomProduit"]=>
    string(18) "Manuel de recettes"
    ["idDeclinaison"]=>
    int(11)
    ["prix"]=>
    string(5) "12.99"
    ["prixSolde"]=>
    NULL
    ["qualite"]=>
    string(6) "Commun"
    ["libelleDeclinaison"]=>
    string(24) "Les Recettes de Flagadus"
    ["descDeclinaison"]=>
    string(37) "Plats délicieux par le chef Flagadus"
  }
}
```

Résultat obtenu si une exception survient (un tableau vide est renvoyé) :

```
array(0) {
}
```

3) Trace côté site (visuelle)

Extrait du code gérant l'erreur :

```
<!-- Section Artefacts Vedettes -->
<section class="w-full py-16 px-4">
  <!-- Si au moins 1 vedette est présente -->
  <?php if (count($vedettes) > 0) : ?>
    <div class="max-w-7xl mx-auto">
      <h2 class="font-display text-4xl text-green-800
text-center mb-12">Artefacts Vedettes</h2>
      <div class="grid grid-cols-1 md:grid-cols-2
lg:grid-cols-4 gap-6" id="featured-products">
        <?php foreach ($vedettes as $prod):
          genererVignetteProduit($prod['idDeclinaison'],
$prod['libelleDeclinaison'], $prod['prix'], $prod['prixSolde'],
$prod['qualite'], $prod['nomProduit'])
        ?>
        <?php endforeach; ?>
      </div>
    </div>
  </section>
```

Résultat obtenu sans exception :



Résultat obtenu si une exception survient (aucun top de produit n'est présent) :



Chargement des catégories enfants d'une catégorie

1) Code

```
/**
 * Fonction permettant de charger les différentes catégories enfants
 * dispos pour une catégorie parent
 */

function chargerCategoriesEnfants($idCategorieParent)
{
    $stab_categories = [];
    try {
        global $connection;
        $statement = $connection->prepare("CALL
getCategEnfants(:idCategorieParent)");
        $statement->bindParam(':idCategorieParent', $idCategorieParent,
PDO::PARAM_INT);
        $statement->execute();

        while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
            $stab_categories[$ligne['idCategorie']] = $ligne['libelle'];
        }

        $statement->closeCursor();
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerCategoriesEnfants : " .
$e->getMessage());
    }

    return $stab_categories;
}
```

2) Trace d'exécution côté php

Cas 1 : ID de la catégorie existant, et des sous catégories sont trouvées

```
//Appel de la fonction

$idCategorieParent = 1; //Catégorie "armes"
$categoriesEnfant = changerCategoriesEnfants($idCategorieParent); //Enfants de
la catégorie "armes"

echo "<pre>";
var_dump($categoriesEnfant);
echo "</pre>";
```

Obtenu :

```
array(6) {
    [3]=>
    string(17) "Dagues & Couteaux"
    [4]=>
    string(7) "Épées"
    [5]=>
    string(6) "Haches"
    [6]=>
    string(8) "Marteaux"
    [7]=>
    string(12) "Armes d'Hast"
    [9]=>
    string(4) "Arcs"
}
```

Cas 2 : ID de la catégorie existant, mais ne possède aucune catégorie enfant (renvoie un tableau vide)

```
//Appel de la fonction

$idCategorieParent = 67; //Catégorie de test n'ayant pas d'enfants
$categoriesEnfant = changerCategoriesEnfants($idCategorieParent);

echo "<pre>";
var_dump($categoriesEnfant);
echo "</pre>";
```


Obtenu :

```
array(0) {  
}
```

Cas 3 : ID de la catégorie inexistant, la procédure lève une exception attrapée, et un tableau vide est renvoyé

```
//Appel de la fonction  
  
$idCategorieParent = 9999; //Catégorie inexistante  
$categoriesEnfant = chargerCategoriesEnfants($idCategorieParent);  
  
echo "<pre>";  
var_dump($categoriesEnfant);  
echo "</pre>";
```

Obtenu :

```
array(0) {  
}
```

3) Trace côté site (visuelle)

Code php / html gérant l'affichage des sous catégories :

```
<!--GESTION DES CATEGORIES----->
        <select name="category" class="px-4 py-2 border-2
border-green-700 rounded-lg focus:outline-none focus:border-yellow-500"
onchange="this.form.submit()">
        <?php
            $selected_category = $category ?? ''; //On la
récup la catégorie si elle existe, sinon on met chaine vide
            echo '<option value="all" selected>Toutes les
catégories</option>'; //Option par défaut
            foreach ($categories as $id => $libelle) {
//Parcours de toutes les catégories disponibles
                $selected = ($selected_category == $id ?
'selected' : ''); //Si la catégorie sélectionnée est la meme que la catégorie
actuelle, on met 'selected'

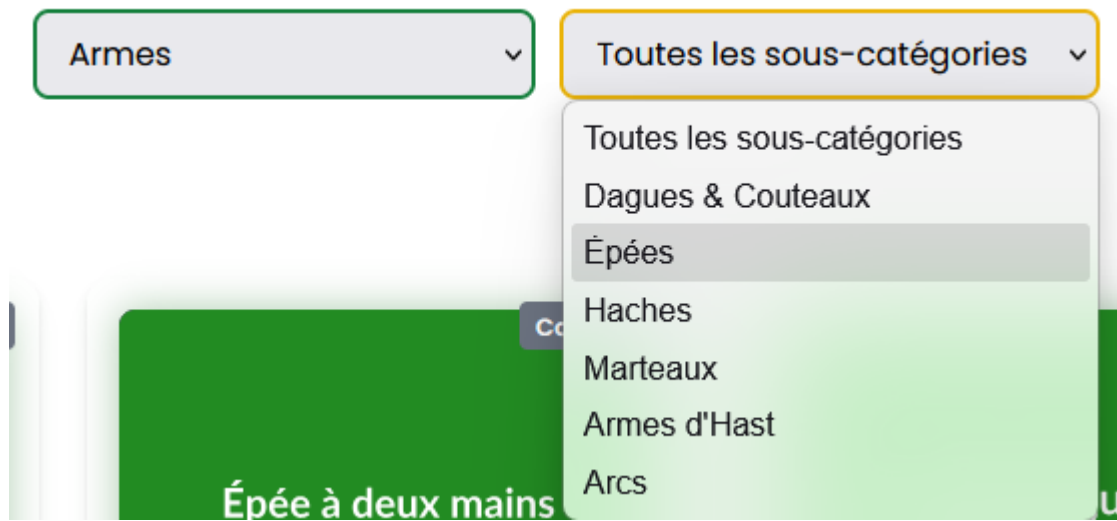
                $selected = htmlentities($selected);

                $id = htmlentities($id);
                $libelle = htmlentities($libelle);
                echo '<option value="' . $id . '"' . $selected
. '>' . $libelle . '</option>'; //On injecte dans une option la catégorie et
si elle est selectionnée select auto
            }
        ?>
    </select>
```

Cas 1 : ID de la catégorie existant, et des sous catégories sont trouvées

Les sous catégories sont bien récupérées par la procédure stockée (on prend juste celles qui ont minimum 1 produit dedans)

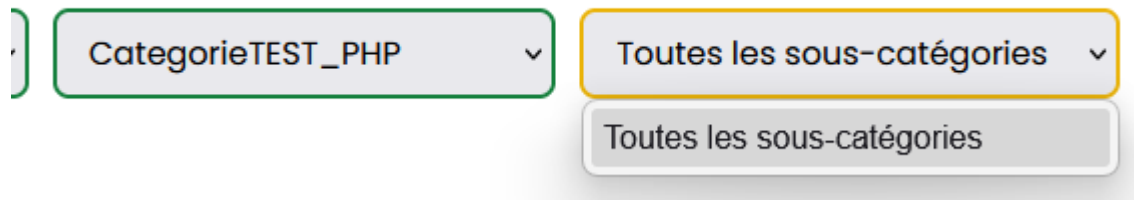
Obtenu :



Cas 2 : ID de la déclinaison existant, mais ne possède aucune catégorie enfant

La catégorie TEST_PHP ne possède aucune catégorie enfant, par défaut on affiche juste "Toutes les sous catégories"

Obtenu :

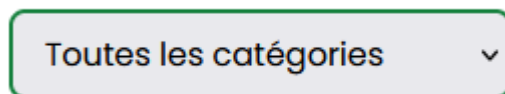


The image shows a web form with two dropdown menus. The first dropdown is labeled "CategorieTEST_PHP" and has a green border. The second dropdown is labeled "Toutes les sous-catégories" and has a yellow border. Below the second dropdown, there is a third dropdown labeled "Toutes les sous-catégories" with a grey border.

Cas 3 : ID de la catégorie inexistant, la procédure lève une exception attrapée, et un tableau vide est renvoyé

On va simuler la soumission la soumission d'une fausse catégorie depuis les paramètres GET en mettant id de la catégorie à 999. On affichera par défaut "Toutes les catégories" si rien n'est trouvé.

Obtenu :



The image shows a single dropdown menu with the label "Toutes les catégories" and a green border.

Chargement de tous les produits d'une catégorie

1) Code

```
/**
 * Récupère tous les id produit d'une catégorie (ainsi que des sous
catégories)
 */
function getProduitsByCategorie($idCategorie)
{
    $produits_categ = [];
    try {
        global $connection;
        // Récupérer tous les produits d'une catégorie (y compris
enfants)
        $statement = $connection->prepare("CALL
getProduitsByCategorie(:idCategorie)");
        $statement->bindParam(':idCategorie', $idCategorie,
PDO::PARAM_INT);
        $statement->execute();
        while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
            $produits_categ[] = $row['idDeclinaison']; // On stocke
tous les id produit de cette catégorie
        }
        $statement->closeCursor();
    } catch (PDOException $e) {
        error_log("Erreur SQL getProduitsByCategorie : " .
$e->getMessage());
    }

    return $produits_categ;
}
```

2) Trace d'exécution côté php

Cas 1 : ID de la catégorie existant, et des produits appartenants à la catégorie sont trouvés

```
//Appel de la fonction

$idCategorieParent = 22; //Catégorie Alchimie et Potions
//Tous les produits dans "Alchimie et Potions" ou dans ses catégs enfant
$produits = getProduitsByCategorie($idCategorieParent);

echo "<pre>";
var_dump($produits);
echo "</pre>";
```

Obtenu : (les id des déclinaisons correspondant à la catégorie choisie)

```
array(8) {
  [0]=>
  int(7)
  [1]=>
  int(8)
  [2]=>
  int(1)
  [3]=>
  int(2)
  [4]=>
  int(3)
  [5]=>
  int(4)
  [6]=>
  int(5)
  [7]=>
  int(6)
}
```

Cas 2 : ID de la catégorie existant, mais ne possède aucun produit à l'intérieur (tableau vide renvoyé)

```
//Appel de la fonction

$idCategorieParent = 67; ////Catégorie de test n'ayant pas de produits
$produits = getProduitsByCategorie($idCategorieParent);

echo "<pre>";
var_dump($produits);
echo "</pre>";
```

Obtenu :

```
array(0) {
}
```

Cas 3 : ID de la catégorie inexistant, exception interceptée et tableau vide renvoyé

```
//Appel de la fonction

$idCategorieParent = 999; ////Catégorie inexistante
$produits = getProduitsByCategorie($idCategorieParent);

echo "<pre>";
var_dump($produits);
echo "</pre>";
```

Obtenu :

```
array(0) {
}
```

3) Trace côté site (visuelle)

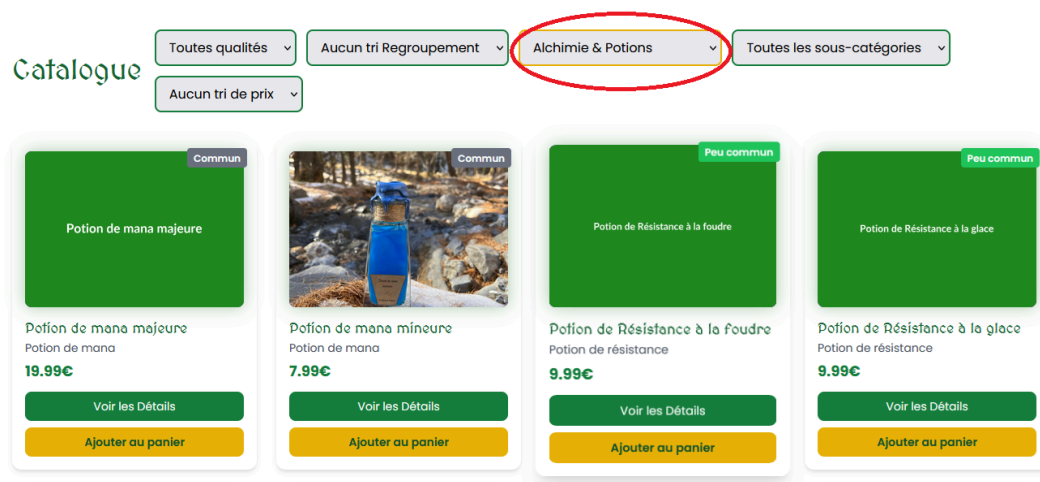
Code php / html gérant l'affichage des produits :

```
<div class="grid grid-cols-1 md:grid-cols-3 lg:grid-cols-4 gap-6 min-h-[30vh]" id="all-products">
    <?php //Chargement des produits correspondants aux critères de recherche
    if (!empty($produits)) { //Si des produits sont trouvés
        foreach ($produits as $cle => $produit) {
            genererVignetteProduit($produit['idDeclinaison'],$produit['libelleDeclinaison'], $produit['prix'], $produit['prixSolde'], $produit['qualite'], $produit['produit_libelle']);?>
        }
    } else { // Aucun produit trouvé ?>
        <p class="mt-8 text-center text-gray-500 text-lg col-span-full w-full">
            Aucun produit ne correspond à votre recherche.
        </p>
    }
    <?php } ?>
</div>
</div>
```

Cas 1 : ID de la catégorie existant, et des produits appartenants à la catégorie sont trouvés

Les produits sont bien chargés et correspondent bien à la catégorie.

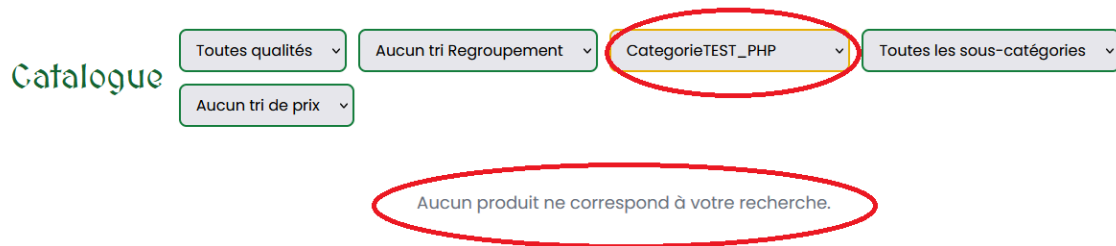
Obtenu :



Cas 2 : ID de la catégorie existant, mais ne possède aucun produit à l'intérieur (tableau vide renvoyé)

La catégorie TEST_PHP ne possède aucun produit, comme vu précédemment un tableau vide est renvoyé donc on affiche "Aucun produit ne correspond à votre recherche."

Obtenu :



Cas 3 : ID de la catégorie inexistant, exception interceptée et tableau vide renvoyé

On va simuler le fait de soumettre une fausse catégorie "999" depuis l'url grâce aux paramètres GET. Aucun produit n'est trouvé, un tableau vide est renvoyé.

Obtenu :



Chargement des produits apparentés à un produit

1) Code

```
/**
 * Récupère dans la BD tous les produits similaires
 */
function chargerProduitsSimilaires($idDeclinaison)
{
    $tab_produitsSimilaires = [];
    try {
        global $connection;
        //On met d'abord les produits apparentés
        $statement = $connection->prepare("CALL
getDeclinaisonsApparente(:idProduit)"); //Récup tous les produits apparentés
        $statement->execute(['idProduit' => $idDeclinaison]); //Executer la
commande
        while ($ligne = $statement->fetch((PDO::FETCH_ASSOC))) { //Récup un
tableau associatif
            $tab_produitsSimilaires[$ligne['idDeclinaison']] = $ligne;
        }
        $statement->closeCursor();
    } catch (PDOException $e) {

        error_log("Erreur SQL chargerProduitsSimilaires : " .
$e->getMessage());
        //en cas d'erreur un tableau vide sera retourné

    }
    return $tab_produitsSimilaires; //On renvoie le tableau des produits
apparentes
}
```

2) Trace d'exécution côté php

Cas 1 : ID de la déclinaison existant, et possède bien des produits apparentés

```
//Appel de la fonction

$idDeclinaison = 1; ////Déclinaison : Potion de mana majeure
$produits_similaires = chargerProduitsSimilaires($idDeclinaison);

echo "<pre>";
var_dump($produits_similaires);
echo "</pre>";
```

Obtenu : (toutes les déclinaisons étant apparentées à la potion de mana majeure)

```
array(15) {
  [3]=>
  array(13) {
    ["idDeclinaison"]=>
    int(3)
    ["libelleDeclinaison"]=>
    string(34) "Potion de Résistance à la foudre"
    ["idProduit"]=>
    int(2)
    ["descDeclinaison"]=>
    string(37) "Résistance aux attaques électriques"
    ["ficheTechnique"]=>
    string(52) "Durée: 3 min, Résistance: +40%, Saveur: Citronnée"
    ["prix"]=>
    string(4) "9.99"
    ["prixSolde"]=>
    NULL
    ["qteStock"]=>
    int(500)
    ["seuilQte"]=>
    int(20)
    ["qualite"]=>
    string(10) "Peu commun"
    ["provenance"]=>
    string(13) "Tour de Babel"
    ["moyenneAvis"]=>
    string(4) "2.00"
    ["libelle"]=>
    string(21) "Potion de résistance"
  }
  [4]=>
  array(13) {
    ["idDeclinaison"]=>
    int(4)
    ["libelleDeclinaison"]=>
    string(33) "Potion de Résistance à la glace"
    .....
  }
```

Cas 2 : ID de la déclinaison existant mais ne possède aucune déclinaison de produit apparentée (tableau vide renvoyé)

```
//Appel de la fonction

$idDeclinaison = 84; ////Déclinaison : Arme de test (apparentée à aucun autre produit)
$produits_similaires = chargerProduitsSimilaires($idDeclinaison);

echo "<pre>";
var_dump($produits_similaires);
echo "</pre>";
```

Obtenu :

```
array(0) {
}
```

Cas 3 : ID de la déclinaison inexistant, exception interceptée et tableau vide renvoyé

```
//Appel de la fonction
$idDeclinaison = 999; ////Déclinaison inexistante
$produits_similaires = chargerProduitsSimilaires($idDeclinaison);
echo "<pre>";
var_dump($produits_similaires);
echo "</pre>";
```

Obtenu :

```
array(0) {
}
```

3) Trace côté site (visuelle)

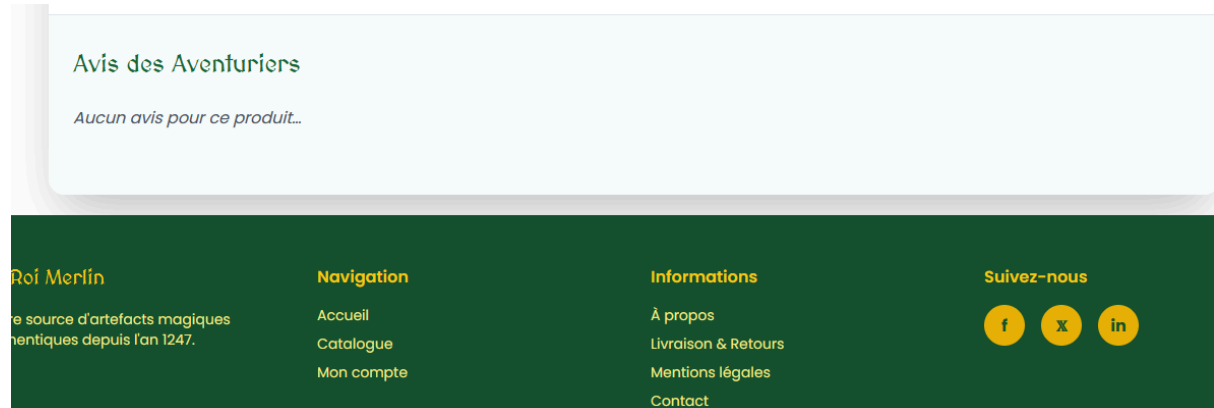
Code php / html gérant l'affichage des produits apparentés :

```
<!-- Produits apparentés -->
    <?php if (!empty($tab_produitsSimilaires)) : ?>
        <div class="border-t px-6 py-8 bg-white">
            <h3 class="font-display text-2xl text-green-800 mb-6">Artefacts Similaires</h3>
            <div class="relative overflow-hidden">
                <div
                    id="related-carousel" class="flex gap-4
transition-transform duration-500 ">
                    <?php
                        //Pour chaque produit apparenté
                        foreach ($tab_produitsSimilaires as $produit) :
                            //Récup les photos du produit apparenté
                            $photos_produits_similaires =
chargerPhotosProduit($produit['idDeclinaison'],
$produit['libelleDeclinaison']);
                            genererVignetteProduit(
                                $produit['idDeclinaison'],
                                $produit['libelleDeclinaison'],
                                $produit['prix'],
                                $produit['prixSolde'],
                                $produit['qualite'],
                                $produit['libelle'],
                                'petite'
                            );
                        ?>
                    <?php endforeach; ?>
                </div>
                <button id="related-prev" class="absolute left-0
top-1/2 transform -translate-y-1/2 bg-white rounded-full p-2 shadow-lg
hover:bg-gray-100 transition-all z-10">
                    <svg class="w-6 h-6 text-green-700" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                        <path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 19l-7-7 7-7" />
                    </svg>
                </button>
```


Cas 2 : ID de la déclinaison existant mais ne possède aucune déclinaison de produit apparentée (tableau vide renvoyé)

La déclinaison ne possède aucune déclinaisons apparentées donc le tableau contenant les éléments apparentés est vide comme vu précédemment. **La section des produits apparentés n'est donc pas affichée.**

Obtenu :



Savoir si un client est autorisé à déposer un avis sur un produit

1) Code

```
//Appel d'une FONCTION stockée

/**
 * Permet de savoir si le client peut déposer un avis, c'est à dire :
 * 1) Le produit à déjà été acheté au moins une fois par un client
 * 2) Le client n'a pas encore déposé d'avis sur le produit (droit à 1
seul avis)
 */
function isAvisAutorise($idClient, $idProduit)
{
    try {
        global $connection;
        // Appel de la fonction stockée
        $stmt = $connection->prepare("SELECT
isProduitAvisPossible(:idClient, :idDeclinaison) AS achete");
        $stmt->execute(['idClient' => $idClient, 'idDeclinaison' =>
$idProduit]);
        $estAchete = $stmt->fetch();
        if ($estAchete['achete'] > 0) { //Si il a bien acheté
            return true;
        }
        return false;
    } catch (PDOException $e) {
        error_log("Erreur SQL isAvisAutorise : " . $e->getMessage());
        //en cas d'erreur on retourne false pour empecher l'avis
        return false;
    }
}
```


2) Trace d'exécution côté php

Cas 1 : ID du client existant et id du produit existant

```
//Appel de la fonction
$idClient = 5;
$idDeclinaison = 62; ////Déclinaison : "Corchets de serrure"
$avisAutorise = isAvisAutorise($idClient,$idDeclinaison);

echo "<pre>";
var_dump($avisAutorise);
echo "</pre>";
```

Obtenu : true car le client a acheté le produit et n'a pas encore déposé d'avis

`bool(true)`

⇒ Si le client **dépose un avis** on obtient `bool(false)` **maintenant:**

⇒ Si le client **n'a jamais acheté** le produit on obtient : `bool(false)`

Cas 2 : ID du client existant et id du produit inexistant (exception interceptée et false renvoyé)

```
//Appel de la fonction
$idClient = 57;
$idDeclinaison = 9999; ////Déclinaison : inexistante
$avisAutorise = isAvisAutorise($idClient,$idDeclinaison);

echo "<pre>";
var_dump($avisAutorise);
echo "</pre>";
```

Obtenu :

`bool(false)`

Cas 3 : ID du client inexistant et id du produit existant (exception interceptée et false renvoyé)

```
//Appel de la fonction
$idClient = 99999; //Client inexistant
$idDeclinaison = 62; ////Déclinaison : Crochets
$avisAutorise = isAvisAutorise($idClient,$idDeclinaison);

echo "<pre>";
var_dump($avisAutorise);
echo "</pre>";
;
```

Obtenu :

```
bool(false)
```

Cas 4 : ID du client inexistant et id du produit inexistant (exception interceptée et false renvoyé)

Obtenu :

```
bool(false)
```

3) Trace côté site (visuelle)

Code php / html gérant l'affichage du formulaire pour déposer un avis :

```
<!-- Formulaire d'avis //Seulement si l'utilisateur est connecté + a acheté
le produit-->
    <?php if (isset($_SESSION["SLeRoiMerlin"]) &&
$_SESSION["SLeRoiMerlin"] == "OK" && isAvisAutorise($_SESSION["idClient"],
$idProduit) == true): ?>
        <div class="bg-white p-6 rounded-lg shadow-md">
            <h4 class="font-semibold text-lg mb-4">Laisser un
avis</h4>

            <form id="review-form" method="post"
action="traitAvis.php" enctype="multipart/form-data">

                <input type="hidden" name="idProduit" value="<?=$
$infos_produit['idDeclinaison'] ?>">
                <div class="mb-4">
                    <div class="mb-4">
```

```

<label for="review-note" class="block
font-semibold mb-2">
    Note (entre 0 et 5) :
</label>

<input
    type="number"
    id="review-note"
    name="note"
    min="0"
    max="5"
    step="1"
    required
    class="w-24 px-4 py-2 border-2
border-gray-300 rounded-lg text-center text-lg
focus:outline-none
focus:border-green-700"
    placeholder="5" />
</div>
</div>
<div class="mb-4">
    <label for="review-text" class="block
font-semibold mb-2">Votre avis (optionnel):</label>
    <textarea id="review-text" name="commentaire"
rows="4" class="w-full px-4 py-2 border-2 border-gray-300 rounded-lg
focus:outline-none focus:border-green-700" placeholder="Partagez votre
expérience..."></textarea>
</div>
<div class="mb-4">
    <label for="review-image" class="block
font-semibold mb-2">Ajouter des images (optionnel, 4 max) :</label>
    <input type="file" id="review-image"
name="imageMedia[]"
    accept="image/jpeg,image/png"
class="w-full px-4 py-2 border-2 border-gray-300 rounded-lg" multiple>
</div>
<button type="submit" name="envoiAvis"
class="bg-green-700 hover:bg-green-800 text-white px-6 py-3 rounded-lg
font-semibold transition-all transform hover:scale-105">
    Publier l'avis
</button>
</form>

```

```
</div>  
<?php endif; ?>
```

Cas 1 : Le client a acheté le produit et n'a jamais déposé d'avis

Le formulaire s'affiche bien, autorisant le client à déposer son avis.

Obtenu :

Avis des Aventuriers

Aucun avis pour ce produit...

Laisser un avis

Note (entre 0 et 5) :

5

Votre avis (optionnel) :

Partagez votre expérience...

Ajouter des images (optionnel, 4 max) :

Parcourir... Aucun fichier sélectionné.

Publier l'avis

Cas 2 : Le client a acheté le produit et a déjà déposé un avis

Le client a déjà déposé son avis, le formulaire ne s'affiche plus.

Obtenu :


Avis des Aventuriers

ROMA Quentin

2025-12-20 13:54:13

★★★★★

Excellent



Cas 3 : Le client n'a jamais acheté le produit

Le formulaire d'avis ne s'affiche pas, empêchant le client non autorisé de déposer un avis.

Obtenu :

Avis des Aventuriers

Aucun avis pour ce produit...

Artefacts Similaires

Cas 4 : Le client n'est pas connecté

Le formulaire d'avis ne s'affiche pas, empêchant le l'individu non connecté de déposer un avis.

Obtenu :

Avis des Aventuriers

Aucun avis pour ce produit...

Artefacts Similaires

Chargement de toutes les infos d'un produit décliné

1) Code

```
/**
 * Récupère dans la BD toutes les infos sur le produit
 */
function chargerProduit($idDeclinaison)
{
    $stab_produit = [];
    try {
        global $connection;
        $statement = $connection->prepare("CALL
getDeclinaisonProduit(:idDeclinaison)"); //Récup les infos sur le
produit
        $statement->execute(['idDeclinaison' => $idDeclinaison]);
        $stab_produit = $statement->fetch(PDO::FETCH_ASSOC); //Récup
un tableau associatif
        $statement->closeCursor();

        return $stab_produit; //On renvoie le tableau des infos du
produit
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerProduit : " . $e->getMessage());
        // retour tab vide si ça plante donc on sera automatiquement
redirigé à index (voir ligne 19)
    }
    return $stab_produit;
}
```

2) Trace d'exécution côté php

Cas 1 : ID de la déclinaison existante

```
//Appel de la fonction
$idDeclinaison = 1; ////Déclinaison : Potion de mana majeure
$declinaison = chargerProduit($idDeclinaison);

echo "<pre>";
var_dump($declinaison);
echo "</pre>";
```

Obtenu : (toutes les informations de la déclinaison)

```
array(14) {
  ["idDeclinaison"]=>
  int(1)
  ["libelleDeclinaison"]=>
  string(22) "Potion de mana majeure"
  ["idProduit"]=>
  int(1)
  ["descDeclinaison"]=>
  string(47) "Régénère une grande quantité de mana (100%)"
  ["ficheTechnique"]=>
  string(51) "Contenance: 200ml, Effet: Instantané, Poids: 0.3kg"
  ["prix"]=>
  string(5) "19.99"
  ["prixSolde"]=>
  NULL
  ["qteStock"]=>
  int(500)
  ["seuilQte"]=>
  int(50)
  ["qualite"]=>
  string(6) "Commun"
  ["provenance"]=>
  string(8) "Falconia"
  ["moyenneAvis"]=>
  string(4) "4.50"
  ["libelle"]=>
  string(14) "Potion de mana"
  ["description"]=>
  string(74) "Fiole contenant un liquide bleu lumineux permettant de restaurer la magie."
}
```

Cas 2 : ID de la déclinaison inexistante, exception interceptée et tableau vide renvoyé

```
//Appel de la fonction
$idDeclinaison = 999999; ////Déclinaison : inexistante
$declinaison = chargerProduit($idDeclinaison);

echo "<pre>";
var_dump($declinaison);
echo "</pre>";
```

Obtenu :

```
array(0) {
}
```

3) Trace côté site (visuelle)

Code php / html gérant l'affichage des produits :

```
<?php
//On essaie de charger le produit
$infos_produit = chargerProduit($idProduit);

if (!$infos_produit) { //Si rien n'a été trouvé sur ce produit (innexistant)
    $erreur = "Produit introuvable.";
    header("Location: catalogue.php?msgErreur=".urlencode($erreur));
    exit;
} ?>

<div id="section-consult-produit" class="w-full py-5 px-4 bg-white">
    <div class="max-w-6xl mx-auto bg-white rounded-2xl shadow-2xl
overflow-y-hidden">

        <!-- En-tête -->
        <div class="sticky top-0 bg-white border-b px-6 py-4 flex
justify-between items-center z-10">
            <h2 id="modal-product-name" class="font-display text-5xl
text-green-800"><?php echo "{$nom_produit}"; ?></h2>
        </div>
        <!-- Contenu de la page..... -->
        <!-- . . . . . -->
        <!-- . . . . . -->
    </div>
</div>
```

Cas 1 : ID de la déclinaison existante

La page du produit est bien affichée, toutes les informations sont présentes.

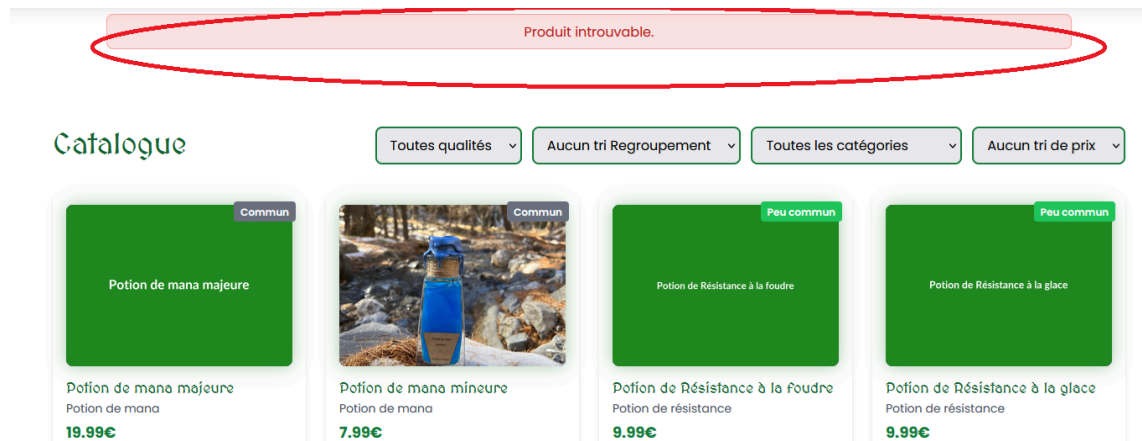
Obtenu :



Cas 2 : ID de la déclinaison inexistant, redirection avec message d'erreur

L'id de la déclinaison chargé est incorrect (paramètre GET modifié dans l'url). Le site redirige automatiquement au catalogue avec un message d'erreur indiquant que le produit n'existe pas.

Obtenu :



Conclusion

Ce livrable a démontré l'utilisation systématique de procédures stockées pour centraliser la logique métier au niveau de la base de données. Les huit procédures présentées couvrent les opérations essentielles du catalogue : chargement des produits, gestion des photos, filtrage par catégories, produits apparentés, et contrôle des autorisations d'avis.

La stratégie de gestion des exceptions mise en place garantit la robustesse de l'application : chaque erreur SQL est interceptée côté PHP et traitée avec un mécanisme de repli approprié (tableaux vides, images placeholder, redirections). Cette approche assure une expérience utilisateur fluide même en cas de données invalides ou absentes.

Les traces d'exécution ont validé le comportement attendu dans tous les scénarios testés : cas nominaux avec données valides, cas limites avec données absentes, et cas d'erreur avec paramètres invalides. L'affichage côté site s'adapte automatiquement à chaque situation grâce aux structures conditionnelles PHP.

Cette architecture procédurale offre des avantages significatifs en termes de maintenance, de sécurité et de performances, tout en facilitant l'évolution future du système d'information.