****

Le site de vente de la sorcière Giselle

Manuel d’utilisation de la Base de Données

**GiselleMagicArts.com**

Afin que la sorcière Giselle puisse utiliser la base de données directement, nous avons recensé dans cette annexe l’ensemble des commandes qu’elle pourrait avoir besoin quotidiennement.

Pour se connecter la sorcière Giselle peut se connecter de deux manières :

* Avec l’utilisateur root : tous les droits accordées
* Avec l’utilisateur giselle : droits restreints à l’affichage des tables et aux procédures stockées

Les mots de passe seront communiqués à la sorcière Giselle de manière sécurisée.

Table des matières

[Affichage du stock des ingrédients et mis à jour à la réception de nouveaux ingrédients : 3](#_Toc479265660)

[Consultation des commandes d’un client : 3](#_Toc479265661)

[Suppression d’un ingrédient arrivé à expiration : 4](#_Toc479265662)

[Mise à jour du stock après envoi d’une commande : 4](#_Toc479265663)

[Ajouter une ou plusieurs nouvelles recettes : 4](#_Toc479265664)

[Valider une recette qui a été acceptée : 5](#_Toc479265665)

[Supprimer une recette qui n’a pas été retenue : 5](#_Toc479265666)

[Les potions ou les onguents réalisables avec un ingrédient précis : 6](#_Toc479265667)

[Les couples (potions, onguents) utilisant la même liste d’ingrédients : 6](#_Toc479265668)

[Les potions réalisables avec un diluant précis : 6](#_Toc479265669)

[La liste des potions classée par température de préparation : 7](#_Toc479265670)

[La marge dégagée pour chaque préparation : 7](#_Toc479265671)

[Le nombre moyen d’ingrédients utilisés pour fabriquer une potion (ou onguent) : 7](#_Toc479265672)

[Recettes enregistrées dans la base de données : 8](#_Toc479265673)

[Ingrédients associés à chaque fournisseur : 8](#_Toc479265674)

[Commandes dont la réception n’a pas encore été confirmée : 8](#_Toc479265675)

[Afficher le contenu d’une commande : 9](#_Toc479265676)

[Afficher la recette d’une potion : 9](#_Toc479265677)

[Ajouter un nouvel ingrédient : 9](#_Toc479265678)

[Vue pour afficher la liste des potions : 9](#_Toc479265679)

[Vue pour afficher la liste des onguents : 10](#_Toc479265680)

[Vue pour afficher les ingrédients qui sont fourni par chaque fournisseur : 10](#_Toc479265681)

[Vue pour afficher la liste des produits avec leurs prix : 10](#_Toc479265682)

[Vue pour afficher les recettes proposées par chaque client : 11](#_Toc479265683)

[Vue pour afficher le nombre moyen d’ingrédients utilisé par potion : 11](#_Toc479265684)

# Affichage du stock des ingrédients et mis à jour à la réception de nouveaux ingrédients :

Afin de mettre à jour le stock des ingrédients on va devoir effectuer la commande avec la syntaxe suivante :

|  |
| --- |
| CALL Affichage\_stocks([”](https://dev.mysql.com/doc/refman/5.7/en/case.html)NomIngredient[”](https://dev.mysql.com/doc/refman/5.7/en/case.html" \o "14.6.5.1 CASE Syntax), QuantiteIngredient, Fraicheur); |

Par exemple :



On demande l’ingrédient « Alexandrite », on veut ajouter une quantité de 3 ingrédients et on souhaite que ces ingrédients disposent d’une fraicheur de 40.

Afin d’afficher le stock des ingrédients on va devoir effectuer la commande précédente mais sans paramètres

Par exemple :



# Consultation des commandes d’un client :

Afin de consulter les commandes d’une personne dans un premier temps nous allons afficher la table « Personne » pour pouvoir consulter les commandes d’une personne existante :



On va devoir effectuer la commande avec la syntaxe suivante pour consulter la commande d’un client défini :

|  |
| --- |
| CALL historique\_cmd(‘NomPersonne’); |

Par exemple :



On veut ici connaitre la commande de la personne possédant le nom de « Naessens »

# Suppression d’un ingrédient arrivé à expiration :

La suppression d’un ingrédient arrivé à expiration est automatisée tous les jours à minuit il est inutile d’effectuer une commande dans le terminal.

# Mise à jour du stock après envoi d’une commande :

Afin de mettre à jour le stock après l’envoi d’une commande on va commencer par afficher la table « Commande » afin de choisir une commande existante, pour cela le statut de la commande doit être en préparation :



On va devoir effectuer la commande avec la syntaxe suivante pour :

|  |
| --- |
| CALL envoi\_commande(NumeroCommande); |

Par exemple :



On veut ici retirer le contenu de la commande « 2 » dans le stock afin de modifier le statut de cette commande

# Ajouter une ou plusieurs nouvelles recettes :

Afin d’ajouter une nouvelle recette on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL ajouter\_proposition(‘NomPotion’, TemperatureDePreparation, ‘NomDiluant’,  ‘NomIngredient1’, QuantiteIngredient1, FraicheurMin1, FraicheurMax1, Temps1,  ‘NomIngredient2’, QuantiteIngredient2, FraicheurMin2, FraicheurMax2, Temps2,  ‘NomIngredient3’, QuantiteIngredient3, FraicheurMin3, FraicheurMax3, Temps3,  ‘NomInventeur’); |

Par exemple :



On veut ici ajouter la potion « Potion de test » avec une température de préparation de 100, avec du « Vin » comme diluant et une liste d’ingrédients précis avec la quantité, la fraicheur et le temps.

# Valider une recette qui a été acceptée :

Afin de valider une proposition on va commencer par afficher les commandes disponibles pour choisir une commande précise :



On va utiliser la syntaxe suivante :

|  |
| --- |
| CALL valider\_proposition(NumeroProposition); |

Par exemple :



On veut ici valider la proposition 4

# Supprimer une recette qui n’a pas été retenue :

Afin de refuser une proposition on va commencer par afficher les commandes disponibles pour choisir une commande précise :



On va utiliser la syntaxe suivante :

|  |
| --- |
| CALL refus\_proposition(4); |

Par exemple :



On veut ici refuser la proposition 4 pour qu’elle soit ensuite supprimée

# Les potions ou les onguents réalisables avec un ingrédient précis :

Afin de consulter les potions ou les onguents qui disposent d’un ingrédient précis on va afficher les ingrédients disponibles afin de choisir un ingrédient précis



On va utiliser la syntaxe suivante :

|  |
| --- |
| CALL preparation\_ingredient(‘NomIngredient’); |

Par exemple :



On veut ici connaitre les potions et les onguents qui disposent de l’ingrédient « Sardonyx »

# Les couples (potions, onguents) utilisant la même liste d’ingrédients :

Afin de consulter les potions et les onguents qui disposent de la même liste d’ingrédients on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL couple\_potion\_onguent(); |

Par exemple :



# Les potions réalisables avec un diluant précis :

Afin de consulter les potions réalisées avec un diluant précis on va afficher les diluants et leurs identifiants :



On va utiliser la syntaxe suivante :

|  |
| --- |
| CALL diluant(‘NomDiluant’); |

Par exemple :



On veut ici connaitre les potions qui utilisent du « Jus de fruit » comme diluant

# La liste des potions classée par température de préparation :

Afin de consulter la liste des potions classée par température on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL temperature(); |

Par exemple :



# La marge dégagée pour chaque préparation :

Afin de consulter la comparaison entre le prix de vente d’une potion et le total du prix des ingrédients on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL marge\_preparation(); |

Par exemple :



# Le nombre moyen d’ingrédients utilisés pour fabriquer une potion (ou onguent) :

Afin de consulter le nombre moyen d’ingrédients utilisés pour fabriquer une potion ou un onguent on va utiliser la syntaxe :

|  |
| --- |
| SELECT \* FROM Moyenne\_ingredient\_potion; |

Par exemple :



# Recettes enregistrées dans la base de données :

Afin de sauvegarder les recettes dans la base de données on va utiliser la syntaxe suivante :

|  |
| --- |
| mysqldump –user=NomUtilisateur –password=PasswordUtilisateur NomBaseDeDonnee NomTable1 NomTable2 > NomDuFichierDeSortie |

Par exemple :



# Ingrédients associés à chaque fournisseur :

Afin de stocker les ingrédients qui sont associés à chaque fournisseur on va utiliser la syntaxe suivante :

|  |
| --- |
| mysqldump –user=NomUtilisateur –password=PasswordUtilisateur NomBaseDeDonnee NomTable1 NomTable2 > NomDuFichierDeSortie |

Par exemple :



# Commandes dont la réception n’a pas encore été confirmée :

Afin de stocker les commandes qui n’ont pas été encore livrées on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL cmd\_livraison(); |

Par exemple :



# Afficher le contenu d’une commande :

Afin d’afficher le contenu d’une commande on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL contenu\_commande(NumeroCommande); |

Par exemple :



# Afficher la recette d’une potion :

Afin d’afficher la recette d’une potion on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL recette\_potion(NumeroProduit); |

Par exemple :



# Ajouter un nouvel ingrédient :

Afin d’ajouter un nouvel ingrédient on va utiliser la syntaxe suivante :

|  |
| --- |
| CALL ajout\_ingredient(‘NomIngredient’, PrixIngredient); |

Par exemple :



# Vue pour afficher la liste des potions :

On va utiliser la syntaxe suivante :

|  |
| --- |
| SELECT \* FROM ListePotion; |

Par exemple :



# Vue pour afficher la liste des onguents :

On va utiliser la syntaxe suivante :

|  |
| --- |
| SELECT \* FROM ListeOnguent; |

Par exemple :



# Vue pour afficher les ingrédients qui sont fourni par chaque fournisseur :

On va utiliser la syntaxe suivante :

|  |
| --- |
| SELECT \* FROM Fournisseur\_produit; |

Par exemple :



# Vue pour afficher la liste des produits avec leurs prix :

On va utiliser la syntaxe suivante :

|  |
| --- |
| SELECT \* FROM Produit\_prix; |

Par exemple :



# Vue pour afficher les recettes proposées par chaque client :

On va utiliser la syntaxe suivante :

|  |
| --- |
| SELECT \* FROM Proposition\_personne; |

Par exemple :



# Vue pour afficher le nombre moyen d’ingrédients utilisé par potion :

On va utiliser la syntaxe suivante :

|  |
| --- |
| SELECT \* FROM Moyenne\_ingredient\_potion; |

Par exemple :

