

GiselleMagicArts.com

Le site de vente de la sorcière Giselle

Manuel d'utilisation de la Base de Données



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 accordés
- 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
Consultation des commandes d'un client :	3
Suppression d'un ingrédient arrivé à expiration :	4
Mise à jour du stock après envoi d'une commande :	4
Ajouter une ou plusieurs nouvelles recettes :	4
Valider une recette qui a été acceptée :	5
Supprimer une recette qui n'a pas été retenue :	5
Les potions ou les onguents réalisables avec un ingrédient précis :	6
Les couples (potions, onguents) utilisant la même liste d'ingrédients :	6
Les potions réalisables avec un diluant précis :	6
La liste des potions classée par température de préparation :	7
La marge dégagée pour chaque préparation :	7
Le nombre moyen d'ingrédients utilisés pour fabriquer une potion (ou onguent) :	7
Recettes enregistrées dans la base de données :	8
Ingrédients associés à chaque fournisseur :	8
Commandes dont la réception n'a pas encore été confirmée :	8
Afficher le contenu d'une commande :	9
Afficher la recette d'une potion :	9
Ajouter un nouvel ingrédient :	9
Vue pour afficher la liste des potions :	9
Vue pour afficher la liste des onguents :	10
Vue pour afficher les ingrédients qui sont fourni par chaque fournisseur :	10
Vue pour afficher la liste des produits avec leurs prix :	10
Vue pour afficher les recettes proposées par chaque client :	11
Vue pour afficher le nombre moyen d'ingrédients utilisé par potion :	11

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("NomIngrédient", QuantiteIngrédient, Fraicheur);
```

Par exemple :

```
mysql> CALL Affichage_stocks("Alexandrite", 3, 40);
```

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

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

Par exemple :

```
mysql> CALL Affichage_stocks("", "", "");
```

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 :

```
mysql> SELECT *  
-> FROM Personne;
```

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

```
CALL historique_cmd('NomPersonne');
```

Par exemple :

```
mysql> call historique_cmd('Naessens');
```

On veut ici connaître 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 :

```
mysql> SELECT *  
-> FROM Commande;
```

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

```
CALL envoi_commande(NumeroCommande);
```

Par exemple :

```
mysql> CALL envoi_commande(2);
```

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 :

```
mysql> CALL ajouter_proposition('Potion de test',100,'Vin','Sardonyx',5,10,5,10,'Azurite',10,5,1,5,'Zircon',1,5,10,2,  
'Merlin');
```

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 fraîcheur 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 :

```
mysql> SELECT *  
-> FROM Proposition;
```

On va utiliser la syntaxe suivante :

```
CALL valider_proposition(NumeroProposition);
```

Par exemple :

```
mysql> CALL valider_proposition(4);
```

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 :

```
mysql> SELECT *  
-> FROM Proposition;
```

On va utiliser la syntaxe suivante :

```
CALL refus_proposition(4);
```

Par exemple :

```
mysql> CALL refus_proposition(4);
```

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

```
mysql> SELECT *  
-> FROM Produit, Ingredient  
-> WHERE Ingredient.IDproduit = Produit.IDProduit;
```

On va utiliser la syntaxe suivante :

```
CALL preparation_ingredient('NomIngredient');
```

Par exemple :

```
mysql> CALL preparation_ingredient('Sardonyx');
```

On veut ici connaître 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 :

```
mysql> CALL couple_potion_onguent();
```

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 :

```
mysql> SELECT IDDiluant, NomDiluant  
-> FROM Diluant;
```

On va utiliser la syntaxe suivante :

```
CALL diluant('NomDiluant');
```

Par exemple :

```
mysql> CALL diluant('Jus de fruit');
```

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 :

```
mysql> CALL temperature();
```

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 :

```
mysql> CALL marge_preparation();
```

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 :

```
mysql> SELECT *  
-> FROM Moyenne_ingredient_potion;
```

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 :

```
mysql> SYSTEM mysqldump --user=pierre --password=pierre  
-> giselle Produit se_compose_prepa Preparation Diluant Ingredient  
-> > tables_produit.sql;
```

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 :

```
mysql> SYSTEM mysqldump --user=pierre --password=pierre  
-> giselle Produit Ingredient Fournisseur fournit  
-> > tables_ingredient_fournisseur.sql;
```

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 :


```
mysql> CALL cmd_livraison();
```

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 :

```
mysql> CALL contenu_commande(2);
```

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 :

```
mysql> CALL recette_potion(2);
```

Ajouter un nouvel ingrédient :

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

```
CALL ajout_ingredient('NomIngredient', PrixIngredient);
```

Par exemple :

```
mysql> CALL ajout_ingredient('Pikachu',100);
```

Vue pour afficher la liste des potions :

On va utiliser la syntaxe suivante :

```
SELECT * FROM ListePotion;
```

Par exemple :

```
mysql> SELECT *  
-> FROM ListePotion;
```

Vue pour afficher la liste des onguents :

On va utiliser la syntaxe suivante :

```
SELECT * FROM ListeOnguent;
```

Par exemple :

```
mysql> SELECT *  
-> FROM ListeOnguent;
```

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

On va utiliser la syntaxe suivante :

```
SELECT * FROM Fournisseur_produit;
```

Par exemple :

```
mysql> SELECT *  
-> FROM Fournisseur_produit;
```

Vue pour afficher la liste des produits avec leurs prix :

On va utiliser la syntaxe suivante :

```
SELECT * FROM Produit_prix;
```

Par exemple :

```
mysql> SELECT *  
-> FROM Produit_prix;
```

Vue pour afficher les recettes proposées par chaque client :

On va utiliser la syntaxe suivante :

```
SELECT * FROM Proposition_personne;
```

Par exemple :

```
mysql> SELECT *  
      -> FROM Proposition_personne;
```

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

On va utiliser la syntaxe suivante :

```
SELECT * FROM Moyenne_ingredient_potion;
```

Par exemple :

```
mysql> SELECT *  
      -> FROM Moyenne_ingredient_potion;
```