

Une erreur est survenue lors du chargement de la version complète de ce site. Veuillez vider le cache de votre navigateur et rafraîchir cette page pour corriger cette erreur.

Les Bases des requêtes SQL

Doctor

Bien le bonjour amis Openiens,

On est là aujourd'hui pour un tutoriel assez simple et rapide. À la demande de quelques uns d'entre vous, on va regarder les requêtes les plus simples.

- INSERT / REPLACE
- DELETE
- SELECT
- UPDATE.

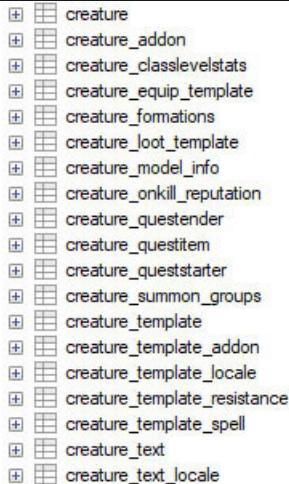
Vous allez voir, c'est vraiment très simple mais ça pourra vous faciliter la vie grandement.

Commençons avec ce que vous savez déjà.

D'après Wikipedia

Une base de données permet de stocker et de retrouver des données structurées, semi-structurées ou des données brutes ou de l'information, souvent en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles.

Pour être plus clair, et pour que ça soit appliqué à WoW, la DB (database) est où on stock toutes les informations que votre client a besoin et qui sont stockées sur le serveur. Prenons l'exemple des créatures.



+	creature
+	creature_addon
+	creature_classlevelstats
+	creature_equip_template
+	creature_formations
+	creature_loot_template
+	creature_model_info
+	creature_onkill_reputation
+	creature_questender
+	creature_questitem
+	creature_queststarter
+	creature_summon_groups
+	creature_template
+	creature_template_addon
+	creature_template_locale
+	creature_template_resistance
+	creature_template_spell
+	creature_text
+	creature_text_locale

La dessus il y a une liste non exhaustive des tables dans lesquelles sont stockées des informations relatives aux créatures.

creature_template : où sont stockées la liste de tous les PNJ. Dedans il y a les informations relatives à leurs points de vie et leur style de combat etc ...

creature : où sont stockés tous les PNJ placés dans les maps, avec les caractéristiques qui ne sont valables qu'à cette créature placée bien précisément. Une entrée dans creature_template peut avoir plusieurs entrées dans la table creature. Mais le contraire ne peut être vrai.

Je ne m'entends pas plus sur le sujet que ça. Passons à la première requête.

INSERT/REPLACE

```
INSERT INTO 'nomdetable' (colonne1, colonne2, colonne3) VALUES (Valeur1,Valeur2,Valeur3);
```

Prenons un exemple un peu plus clair pour nous.

Nous allons créer un PNJ ensemble avec le minimum de colonnes. Sachez que nous ne sommes pas obligés de citer toutes les colonnes mais

seulement celles dont on a besoin de modifié. Les autres auront les valeurs par défaut.

```
INSERT INTO 'creature_template' (entry, modelid1, name, subname, minlevel, maxlevel, unit_class) VALUES (5000, 18, 'Vieux Berger', 'Foret d'Elwynn', 15, 15, 1);
```

Ce qu'on vient de créer c'est un PNJ qui se nomme "Vieux Berger" <Foret d'Elwynn>. Son Id = 5000, et son modelID (l'ID de son physique) = 18. Son niveau est compris entre 15-15 et c'est un guerrier.

Le replaces fonctionne exactement de la même manière :

```
REPLACE INTO 'nomdetable' (colonne1, colonne2, colonne3) VALUES (Valeur1,Valeur2,Valeur3);
```

à la différence que lui si il y a déjà un ligne avec la même entry, il va la supprimer et la remplacer avec la nouvelles. **Faites bien attention à l'utiliser et ne le faite que si vous savez ce que vous faites.**

DELETE

Celui là est encore plus simple.

```
DELETE FROM 'nomdetable' WHERE 'colonneX'='valeurX';
```

Voyons voir un peu dans un cas un peu plus pratiques.

Je viens de créer mon petit Berger, et je l'ai spawn un peu partout dans les maps. Maintenant je le regrette, comment faire.

```
DELETE FROM 'creature' WHERE id =5000;
```

Si je par exemple je ne veux que supprimer les spawns de mon berger à Kalimdor et laissez les autres zones c'est aussi possible.

```
DELETE FROM 'creature' WHERE id =5000 AND map =1;
```

SELECT

Le select est plus utile pour visualiser des informations stocker dans la base de données et fonctionne de cette manières.

```
SELECT FROM 'nomdetable' WHERE 'colonneX' = ValeurX;
```

Par exemple si je veux voir tous les PNJ qui ressemblent traits pour traits à mon Berger.

```
SELECT * FROM 'creature_template' WHERE 'modelid1'=18;
```

Le résultat là c'est que tu auras toutes les informations autour de tous les PNJ utilisant ce modelID.

Si on veut avoir seulement leur nom on peut le faire autrement.

```
SELECT name FROM 'creature_template' WHERE 'modelid1'=18;
```

On peut faire des recherches d'une manière différentes.

```
SELECT * FROM 'creature_template' WHERE 'entry'>18;
```

```
SELECT * FROM 'creature_template' WHERE 'entry' IN (5,4,67,98);
```

A la place du = on peut mettre ">, <, **IN**, **BETWEEN**".

UPDATE

Cette requête est celle que vous allez utilisé **le plus**.

```
UPDATE 'nomdetable' SET 'colonne4' = 'valeur4' WHERE 'colonne1' = 'valeur1';
```

Pour notre cas, on va demander à ce que **tous les berger placés dans** les maps se déplacent dans un radius de 5m.

```
UPDATE 'creature' SET MovementType = 1, wander_distance = 5 WHERE id = 5000;
```

Toutes les entrées dans la table creature ayant comme id 5000 auront un MovementType = 1 Wanderer, et une wander_distance de 5m.

Voilà j'ai essayé de faire quelques choses assez rapide et bien. J'espère que ça vous aiderait.

Thorgrim

Merci énormément pour ton tutoriel, je pense qu'il pourrait vraiment être pas mal de travailler ensemble sur un guide sur le SQL comme je l'ai pour le Lua ☐

Est-ce que ça te tente ?

Doctor

Avec grand plaisir !
