

Module PHP

Cours 6

SOMMAIRE

- Les jointures
- Les expressions régulières (regex)
- Htaccess
- Gestion des droits
- Exercices

LES JOINTURES

Nous avons vu que pour retrouver les données d'une table, on utilise une instruction SQL **SELECT**. Elle est divisée en liste de sélection (la partie qui liste les colonnes à retourner), une liste de tables (la partie qui liste les tables à partir desquelles les données seront retrouvées) et une qualification optionnelle (la partie qui spécifie les restrictions). Par exemple, pour retrouver toutes les lignes de la table `jeux_video`:

```
SELECT * FROM jeux_video;
```

Nous pouvons demander à ce que les résultats d'une requête soient renvoyés dans un ordre trié :

```
SELECT * FROM jeux_video ORDER BY catégorie;
```

LES JOINTURES

Jusqu'ici, nos requêtes avaient seulement consulté une table à la fois. Les requêtes peuvent accéder à plusieurs tables en même temps ou accéder à la même table de façon à ce que plusieurs lignes de la table soient traitées en même temps.

Alors qu'est ce qu'une jointure?

Une requête qui consulte plusieurs lignes de la même ou de différentes tables en même temps est appelée requête de *jointure*.

Exemple, supposons que nous souhaitons comparer la colonne *id* de chaque ligne de la table *users* avec la colonne *id_users* de toutes les lignes de la table *connexions* et que vous choisissiez les paires de lignes où ces valeurs correspondent.

En d'autres termes, une jointure permet d'extraire les données de plusieurs tables à travers une requête qui lie les tables par le biais d'un identifiant commun.

LES JOINTURES

Comment ça marche?

```
SELECT table1.colonne1, table1.colonne2, table1.colonne3, table1.colonne4,  
       table2.colonne3, FROM table1, table2 WHERE table1.colonne2= table2.colonne3;
```

Il est possible d'écrire cette requête autrement:

```
SELECT * FROM table1 INNER JOIN table2 ON (table1.colonne2= table2.colonne3);
```

Il est possible d'indiquer le nombre de lignes affichées sur la première page du résultat en utilisant **LIMIT** dans la requête, par défaut, 30 résultats sont affichés, si vous en avez plus il faudra naviguer entre les pages.

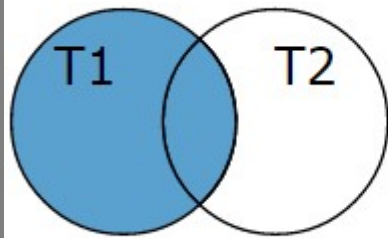
Exemple:

```
SELECT * FROM table1 INNER JOIN table2 ON (table1.colonne2=  
       table2.colonne3) LIMIT 0,25;
```

LES JOINTURES

Les autres types de jointures:

Left Join

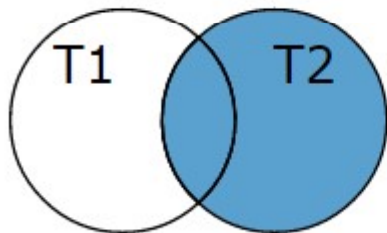


SQL

```
1 | SELECT *  
2 | FROM T1  
3 | LEFT JOIN T2  
4 | ON T1.auteur_id = T2.id
```

LES JOINTURES

Right Join:

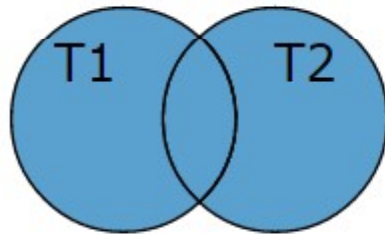


SQL

```
1 | SELECT *  
2 | FROM T1  
3 | RIGHT JOIN T2  
4 | ON T1.auteur_id = T2.id
```

LES JOINTURES

Full Join:

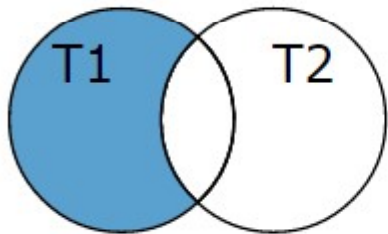


SQL

```
1 SELECT *  
2 FROM T1  
3 FULL JOIN T2  
4 ON T1.auteur_id = T2.id
```


LES JOINTURES

Left Join (sans correspondance):



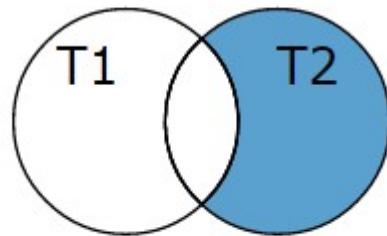
SQL

```
1 SELECT *  
2 FROM T1  
3 LEFT JOIN T2  
4 ON T1.auteur_id = T2.id  
5 WHERE T1.auteur_id IS NULL
```

LES JOINTURES

Right Join (sans correspondance):

Right Join (sans correspondance)

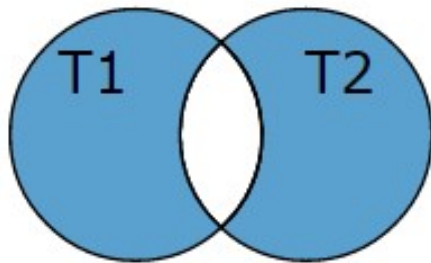


SQL

```
1 | SELECT *  
2 | FROM T1  
3 | RIGHT JOIN T2  
4 | ON T1.auteur_id = T2.id  
5 | WHERE T1.auteur_id IS NULL
```

LES JOINTURES

Full Join (sans correspondance):



SQL

```
1 | SELECT *  
2 | FROM T1  
3 | FULL JOIN T2  
4 | ON T1.auteur_id = T2.id  
5 | WHERE T2.id IS NULL  
6 | OR T1.auteur_id IS NULL
```

LES EXPRESSIONS RÉGULIÈRES (REGEX)

Qu'est ce que c'est?

À quoi ça sert?

Les expressions régulières permettent de trouver, d'identifier ou de remplacer un mot, un caractère ou tout autre type de chaîne.

LES EXPRESSIONS RÉGULIÈRES (REGEX)

Comment ça marche?

Voyons un aperçu des fonctions de filtrage intégrées de PHP les plus utilisées avant d'aller plus loin.

`preg_match()` = recherche un motif dans la chaîne de caractères, retourne true si le motif existe, et false sinon.

`preg_match_all()` = Effectue une recherche globale sur toutes les occurrences du motif dans la chaîne. Utile pour la recherche et le remplacement.

`preg_replace()` = Effectue une recherche et un remplacement d'expressions régulières.

`preg_grep()` = recherche tous les éléments du tableau d'entrée, retournant tous les éléments correspondant au motif regex dans une chaîne.

`preg_split()` = Divise une chaîne de caractères en chaînes de caractères en utilisant une expression régulière.

`preg_quote()` = Cite les caractères d'expression régulière trouvés dans une chaîne.

À noter que la fonction PHP `preg_match()` arrête la recherche après avoir trouvé la première correspondance, alors que la fonction `preg_match_all()` continue la recherche jusqu'à la fin de la chaîne et trouve toutes les correspondances possibles au lieu de s'arrêter au premier résultat.

LES EXPRESSIONS RÉGULIÈRES (REGEX)

RegEx	fonction
.	Correspond à n'importe quel caractère simple sauf le caractère de nouvelle ligne \n.
test	Correspond à la chaîne "test".
^test	Correspond à la chaîne "test" en début de chaîne.
test\$	Correspond à la chaîne "test" en fin de chaîne.
^test\$	Correspond exactement à la chaîne « test ».
[abc]	Correspond à l'un des caractères a, b ou c.
[^abc]	Correspond à un caractère autre que a, b ou c.
[a-z]	Correspond à n'importe quel caractère, de a minuscule à z minuscule
[^A-Z]	Correspond à n'importe quel caractère autre qu'une majuscule
[a-z]+	Correspond à un ou plus caractères minuscules
[0-9.-]	Correspond à n'importe quel nombre, point ou tiret

LES EXPRESSIONS RÉGULIÈRES (REGEX)

RegEx	fonction
<code>\d</code>	Correspond à n'importe quel chiffre décimal. Équivalent à <code>[0-9]</code> .
<code>\D</code>	Correspond à n'importe quel caractère non numérique. Équivalent à <code>^[^0-9]</code>
<code>\s</code>	Correspond à n'importe quel caractère d'espacement (espace, tabulation, nouvelle ligne ou caractère de retour chariot). Similaire à <code>[\t\n\r]</code>
<code>\S</code>	Correspond à n'importe quel caractère qui n'est pas un caractère d'espacement
<code>\w</code>	Correspond à n'importe quel caractère de mot (défini de a à z, de A à Z, de 0 à 9, et le trait de soulignement). Identique à <code>[a-zA-Z_0-9]</code>
<code>\W</code>	Correspond à n'importe quel caractère qui n'est pas un mot. Identique à <code>^[^a-zA-Z_0-9]</code>
<code>(gif jpg)</code>	Correspond à "gif" ou "jpg"
<code>^[a-zA-Z0-9_]{1,}\$</code>	Correspond à un mot d'au moins une lettre, un chiffre ou "_"
<code>([wx])([yz])</code>	Correspond à "wy", "wz", "xy" ou "xz"
<code>[^A-Za-z0-9]</code>	Correspond à un symbole (ni une lettre, ni un nombre)
<code>([A-Z]{3} [0-9]{4})</code>	Correspond à 3 lettres ou 4 nombres

LES EXPRESSIONS RÉGULIÈRES (REGEX)

Comment ça marche?

Exemple:

- Voici comment vérifier le format d'une adresse mail:

```
$email = "contact@exemple.com";  
if (preg_match('/[-0-9a-zA-Z._]+@[-0-9a-zA-Z._]+.[a-zA-Z]{2,4}/igm',  
    $email)) {  
    echo "l'mail est ok.";  
}  
else {  
    echo "erreur sur le mail.";  
}
```


LES EXPRESSIONS RÉGULIÈRES (REGEX)

Comment ça marche?

Exemple:

- Voici comment vérifier le format d'un numéro de téléphone:

```
$tel = '+331000000000';
```

```
if (preg_match('/((\+)33|0|0033)[1-9](\d{2}){4}/igm', $tel)) {  
    echo "Your tel is ok.";  
}  
else {  
    echo "Wrong tel.";  
}
```

HTACCESS

Qu'est ce que c'est?

Htaccess est l'abréviation de hypertext access. C'est un fichier de Configuration utilisé par les serveurs web Apache. Il permet de configurer les paramètres initiaux d'un programme, ou dans le serveur. le fichier .htaccess peut être utilisé pour régler le fonctionnement du serveur. Chaque fonction est une ligne de texte ou un code qui indique des commandes au serveur. Vous pouvez ajouter ou modifier la fonctionnalité en ajoutant ou modifiant le code dans le fichier .htaccess.

À quoi ça sert?

Le htaccess sera utile pour, par exemple:

- Protéger votre site avec un mot de passe.
- Créer une page d'erreur personnalisée.
- Rediriger les visiteurs vers une autre page.
- Ré-écriture des URL

HTACCESS

Comment ça marche?

Le fichier .htaccess commence avec un point.

Voyons comment protéger un dossier avec un fichier .htaccess:

<https://www.seb-huss.com/informatique/php/creer-un-fichier-htaccess-pour-proteger-un-dossier/>

GESTION DES DROITS

La gestion de droits c'est le fait d'autoriser des actions à un type d'utilisateur donné.

Par exemple l'administrateur pour supprimer un compte alors qu'un simple utilisateur n'aura pas ce droit.

Une personne possédant un compte pourra avoir accès à son profil, il pourra gérer un certain nombre d'éléments.

Alors comment peut-on gérer cela? Qu'est ce qui vous vient à l'idée? Comment peut-on identifier qui aura quel(s) rôle(s)?

EXERCICES

- Sur le formulaire d'inscription, vérifier que le code postal est au bon format à l'aide des regEx.
- Créer un fichier .htaccess protégeant un dossier par mot de passe.
- Dans votre base de données assurez-vous que les utilisateurs ont un compte simple, puis créé un seul compte admin. Lorsque l'admin se connecte à son profil, ce dernier doit voir un titre indiquant son rôle.