

Accueil > Cours > Concevez votre site web avec PHP et MySQL > Les fonctions SQL

Concevez votre site web avec PHP et MySQL

🕒 70 heures 📊 Moyenne

Mis à jour le 21/05/2021



Les fonctions SQL

🔒 [Connectez-vous](#) ou [inscrivez-vous](#) gratuitement pour bénéficier de toutes les fonctionnalités de ce cours !



Vous connaissez déjà les fonctions en PHP, mais vous allez découvrir dans ce chapitre que SQL propose lui aussi toute une série de fonctions ! Le langage SQL permet en effet d'effectuer des calculs directement sur ses données, à l'aide de fonctions toutes prêtes.

Celles-ci sont moins nombreuses qu'en PHP, mais elles sont spécialement dédiées aux bases de données et se révèlent très puissantes dans la pratique. Pour reprendre notre exemple de la table

`jeux_video` , elles permettent de récupérer très simplement le prix moyen de l'ensemble des jeux, de compter le nombre de jeux que possède chaque personne, d'extraire le jeu le plus cher ou le moins cher, etc. Les fonctions se révèlent également indispensables lorsqu'on doit travailler avec des dates en SQL, comme nous le ferons dans le chapitre suivant.

Les fonctions SQL peuvent être classées en deux catégories :

- **les fonctions scalaires** : elles agissent sur chaque entrée. Par exemple, vous pouvez transformer en majuscules la valeur de chacune des entrées d'un champ ;
- **les fonctions d'agrégat** : lorsque vous utilisez ce type de fonctions, des calculs sont faits sur l'ensemble de la table pour retourner **une** valeur. Par exemple, calculer la moyenne des prix retourne une valeur : le prix moyen.

Les fonctions scalaires



Nous allons d'abord découvrir le mode d'emploi d'une fonction SQL de type **scalaire** : la fonction `UPPER` . Lorsque vous aurez appris à vous en servir, vous serez capable de faire de même avec toutes les autres fonctions scalaires. Je vous proposerai alors une petite sélection de fonctions scalaires à connaître, sachant qu'il en existe d'autres mais que nous ne pouvons pas toutes les passer en revue, car ce serait bien trop long.

Utiliser une fonction scalaire SQL

Pour nos exemples, nous allons nous baser sur la table `jeux_video` que nous connaissons bien maintenant. Pour rappel, voici à quoi elle ressemble :

| ID | nom | possesseur | console | prix | nbre_joueurs_max | commentaires |
|----|-------------------------|------------|-------------|------|------------------|--|
| 1 | Super Mario Bros | Florent | NES | 4 | 1 | Un jeu d'anthologie ! |
| 2 | Sonic | Patrick | Mega Drive | 2 | 1 | Pour moi, le meilleur jeu au monde ! |
| 3 | Zelda : Ocarina of Time | Florent | Nintendo 64 | 15 | 1 | Un jeu grand, beau et complet, comme on en voit rarement de nos jours. |
| 4 | Mario Kart 64 | Florent | Nintendo 64 | 25 | 4 | Un excellent jeu de kart ! |

| ID | nom | possesseur | console | prix | nbre_joueurs_max | commentaires |
|----|------------------------|------------|----------|------|------------------|-----------------------------|
| 5 | Super Smash Bros Melee | Michel | GameCube | 55 | 4 | Un jeu de baston délirant ! |

On écrit les noms des fonctions SQL en majuscules, comme on le fait déjà pour la plupart des mots-clés comme `SELECT` , `INSERT` , etc. Ce n'est pas une obligation mais plutôt une convention, une habitude qu'ont prise les programmeurs.

Pour vous montrer comment on utilise les fonctions scalaires SQL, je vais me baser sur la fonction `UPPER()` qui permet de convertir l'intégralité d'un champ en majuscules. Supposons que nous souhaitons obtenir les noms de tous les jeux en majuscules ; voici comment on écrirait la requête SQL :

php

```
1 SELECT UPPER(nom) FROM jeux_video
```

La fonction `UPPER` est utilisée sur le champ `nom` . On récupère ainsi tous les noms des jeux en majuscules.



Cela modifie-t-il le contenu de la table ?

Non ! La table reste la même. La fonction `UPPER` modifie seulement la valeur envoyée à PHP. On ne touche donc pas au contenu de la table.

Cela crée en fait un « champ virtuel » qui n'existe que le temps de la requête. Il est conseillé de donner un nom à ce champ virtuel qui représente les noms des jeux en majuscules. Il faut utiliser pour cela le mot-clé `AS` , comme ceci :

php

```
1 SELECT UPPER(nom) AS nom_maj FROM jeux_video
```

On récupère les noms des jeux en majuscules via un champ virtuel appelé `nom_maj` .



Ce champ virtuel est appelé **alias**.

Voici le tableau que retournera MySQL après la requête précédente :

| nom_maj |
|------------------|
| SUPER MARIO BROS |

| nom_maj |
|-------------------------|
| SONIC |
| ZELDA : OCARINA OF TIME |
| MARIO KART 64 |
| SUPER SMASH BROS MELEE |

On peut s'en servir en PHP pour afficher les noms des jeux en majuscules :

php

```
1 <?php
2 $reponse = $bdd->query('SELECT UPPER(nom) AS nom_maj FROM jeux_video');
3
4 while ($donnees = $reponse->fetch())
5 {
6     echo $donnees['nom_maj'] . '<br />';
7 }
8
9 $reponse->closeCursor();
10
11 ?>
```

SUPER MARIO BROS
SONIC
ZELDA : OCARINA OF TIME
MARIO KART 64
SUPER SMASH BROS MELEE
DEAD OR ALIVE
DEAD OR ALIVE XTREME BEACH VOLLEY BALL
ENTER THE MATRIX
MAX PAYNE 2
YOSHI'S ISLAND
COMMANDOS 3
FINAL FANTASY X
POKEMON RUBIS
STARCRAFT
GRAND THEFT AUTO 3
HOMEWORLD 2
ALADIN
SUPER MARIO BROS 3
SSX 3
STAR WARS : JEDI OUTCAST
ACTUA SOCCER 3
TIME CRISIS 3
X-FILES

Afficher les noms des jeux en majuscules

Comme vous le voyez, PHP ne récupère qu'un champ nommé `nom_maj` (même s'il n'existe pas dans la table). En affichant le contenu de ce champ, on ne récupère que les noms des jeux en majuscules.

Bien entendu, vous pouvez aussi récupérer le contenu des autres champs comme avant, sans forcément leur appliquer une fonction :

php

```
1 SELECT UPPER(nom) AS nom_maj, possesseur, console, prix FROM jeux_video
```

On récupérera alors les données suivantes :

| nom_maj | possesseur | console | prix |
|---------|------------|---------|------|
| | | | |

| nom_maj | possesseur | console | prix |
|-------------------------|------------|-------------|------|
| SUPER MARIO BROS | Florent | NES | 4 |
| SONIC | Patrick | Mega Drive | 2 |
| ZELDA : OCARINA OF TIME | Florent | Nintendo 64 | 15 |
| MARIO KART 64 | Florent | Nintendo 64 | 25 |
| SUPER SMASH BROS MELEE | Michel | GameCube | 55 |

Vous savez maintenant utiliser une fonction SQL scalaire. ;-)

Passons en revue quelques fonctions du même type, et qui s'utilisent donc de la même manière.

Présentation de quelques fonctions scalaires utiles

Je vais vous présenter une sélection de fonctions scalaires qu'il peut être utile de connaître. Il en existe bien d'autres, comme nous le verrons à la fin de cette liste, mais il serait trop long et peu utile de toutes les présenter ici.

UPPER : convertir en majuscules

Cette fonction convertit le texte d'un champ en majuscules. Nous l'avons découverte pour introduire les fonctions SQL :

php

```
1 SELECT UPPER(nom) AS nom_maj FROM jeux_video
```

Ainsi, le jeu « Sonic » sera renvoyé sous la forme « SONIC » dans un champ nommé `nom_maj` .

LOWER : convertir en minuscules

Cette fonction a l'effet inverse : le contenu sera entièrement écrit en minuscules.

php

```
1 SELECT LOWER(nom) AS nom_min FROM jeux_video
```

Cette fois, le jeu « Sonic » sera renvoyé sous la forme « sonic » dans un champ nommé `nom_min` .

LENGTH : compter le nombre de caractères

Vous pouvez obtenir la longueur d'un champ avec la fonction `LENGTH()` :

php

```
1 SELECT LENGTH(nom) AS longueur_nom FROM jeux_video
```

Pour « Sonic », on récupérera donc la valeur 5 dans un champ `longueur_nom` .

ROUND : arrondir un nombre décimal

La fonction `ROUND()` s'utilise sur des champs comportant des valeurs décimales. Il n'y en a pas dans la table `jeux_video`, mais si on avait des prix décimaux, on pourrait arrondir les valeurs avec cette fonction.

Celle-ci prend cette fois deux paramètres : le nom du champ à arrondir et le nombre de chiffres après la virgule que l'on souhaite obtenir. Exemple :

php

```
1 SELECT ROUND(prix, 2) AS prix_arrondi FROM jeux_video
```

Ainsi, si un jeu coûte 25,86999 euros, on obtiendra la valeur 25,87 euros dans un champ `prix_arrondi`.

Et bien d'autres !

Il existe beaucoup d'autres fonctions SQL du même type, mais je ne peux pas toutes vous les présenter. La documentation de MySQL vous propose [une liste bien plus complète de fonctions mathématiques](#) (comme `ROUND`) et de [fonctions sur les chaînes de caractères](#) (comme `UPPER`). Si vous voulez en découvrir d'autres, c'est par là qu'il faut aller !

Les fonctions d'agrégat



GROUP BY et HAVING : le groupement de données



TP : UN MINICHAT

LES DATES EN SQL



Les professeurs



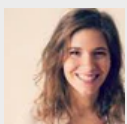
Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)



Mickaël Andrieu

Consultant Web et Data, aka "Individu multi potentiel et turbulent" 😊



Laurène Castor

Learning Designer, Enseignante et Mentor chez OpenClassrooms

Découvrez aussi ce cours en...



Livre



PDF

OPENCLASSROOMS



POUR LES ENTREPRISES



AIDE



EN PLUS



Français



Télécharger dans
l'App Store