

# Interfaçage Flash-PHP-MySQL

---

Après l'interfaçage Flash-PHP-Txt (voir chapitre 11), nous vous proposons de découvrir l'interfaçage Flash-PHP-MySQL. Toujours par l'intermédiaire de scripts PHP, il permet d'ajouter ou d'actualiser des informations dans une base de données MySQL depuis une animation Flash ou encore de récupérer et de mettre en forme des informations issues de cette même base.

Ce chapitre sera découpé en deux grandes parties. Dans la première, nous présenterons les techniques pour interfacier des scripts PHP avec une base MySQL. Dans la seconde, nous mettrons en application ces techniques pour réaliser des interfaçages Flash-MySQL.

## Interfaçage PHP-MySql

Dans les chapitres précédents, nous avons manipulé les informations de la base de données MySQL par l'intermédiaire du gestionnaire phpMyAdmin. Cependant, malgré son interface de gestion assez simple, cet outil est plutôt réservé aux développeurs.

Afin que les utilisateurs puissent interagir sur la base, il leur faut développer des interfaces de gestion de données sur mesure et sécurisées à l'aide de fonctions PHP dédiées à MySQL et de requêtes SQL.

Ces interfaces sont en général matérialisées par des scripts PHP auxquels on envoie des variables depuis un simple formulaire ou une animation Flash. La première fonction de ces scripts est d'établir une liaison avec la base de données (la « connexion »), puis d'élaborer une requête SQL à partir des variables reçues et de transmettre cette requête à la base de données. Après réception de cette requête, la base de données recherche et met en forme les informations demandées et les retourne aux scripts émetteurs. La seconde fonction des scripts est d'informer l'utilisateur des résultats en les affichant à l'écran ou en les retournant à l'animation Flash qui est à l'origine de la demande.

## Concept de la connexion à la base

Pour créer une interface de gestion de données, il faut configurer une connexion à la base de données afin que les scripts PHP soient authentifiés comme ayant un droit d'accès aux données de la base ciblée.

Pour que vous compreniez bien le concept de la connexion à la base, nous allons comparer cette connexion à l'ouverture d'un « canal » entre la base et les scripts dynamiques du site (voir figure 18-1). Ce canal permet d'accéder aux différents champs qui constituent les tables de la base à l'aide de requêtes SQL puis de récupérer une sélection d'enregistrements.

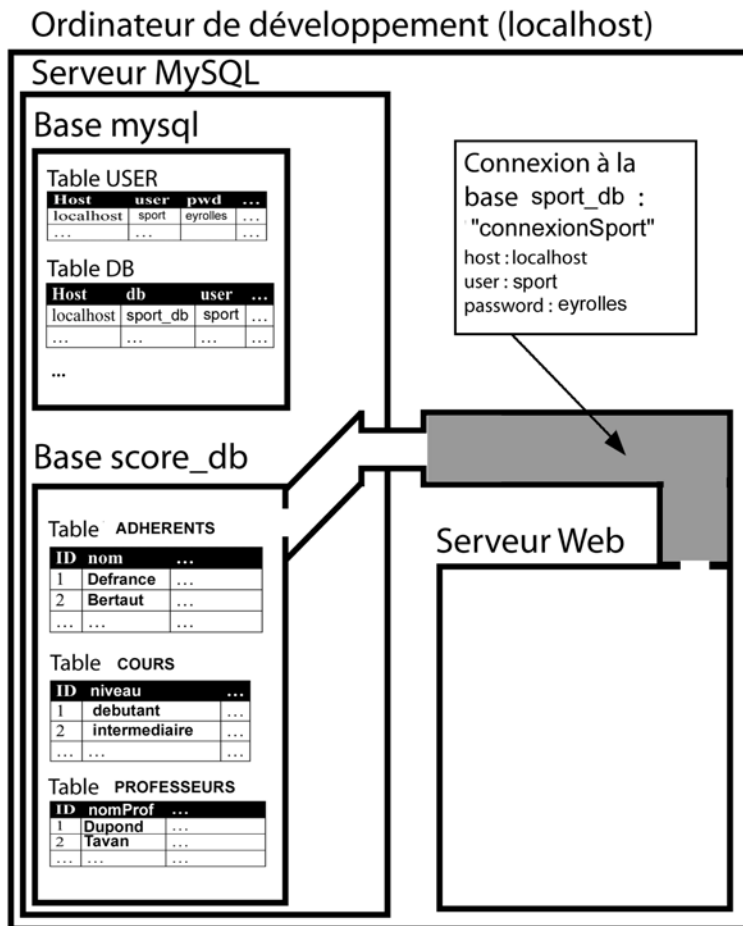


Figure 18-1

*Schéma de principe d'une connexion à la base de données : la connexion peut être comparée à un « canal » autorisant le dialogue entre le serveur Web et le serveur MySQL.*

## Création d'un fichier de connexion

### Alternative au fichier de connexion

La création d'un fichier de connexion à la base de données est intéressante dès que plusieurs scripts d'un même site doivent s'y connecter. Les paramètres de configuration au serveur MySQL sont ainsi regroupés dans un même fichier, ce qui facilite leur modification lors du transfert du site sur un serveur différent. Néanmoins, si votre application dynamique est la seule à nécessiter un accès à la base de données, vous pouvez aussi insérer les paramètres et instructions de connexion directement dans le code de l'application en utilisant le script ci-dessous (vous trouverez un exemple illustrant cette technique dans le fichier `signetsXml.php` de l'application des signets dynamiques du chapitre 22).

```
//-----Configuration isolée de la connexion MySQL
mysql_connect(localhost, sport, eyrolles);
mysql_select_db(sport_db);
```

La connexion à la base utilise des fonctions dédiées à la gestion de MySQL (en général, ces fonctions commencent par le préfixe `mysql_`, par exemple `mysql_connect` ou `mysql_select_db`). Les paramètres nécessaires à la connexion (identifiant, mot de passe, etc.) sont souvent regroupés dans un fichier externe afin de faciliter sa maintenance en cas de modification. Ce fichier de connexion peut être réalisé en mode code dans un simple éditeur ou en mode assisté à l'aide d'une boîte de dialogue de l'interface de Dreamweaver. La deuxième solution permet de disposer d'un fichier de connexion compatible avec la création de pages dynamiques générées par des comportements serveur de Dreamweaver. Vous pourrez ainsi utiliser le même fichier de connexion pour vos applications Flash et pour les pages dynamiques du site.

Afin que vous puissiez choisir la solution qui vous convient le mieux, nous détaillerons ces deux solutions ci-dessous.

Dans le chapitre 16 dédié au gestionnaire de base de données phpMyAdmin, nous avons créé un compte utilisateur `sport` dans la table `user` de la base `mysql`. Nous reprendrons ces mêmes paramètres ci-dessous afin d'être reconnu par le serveur MySQL comme un utilisateur de la base `sport_db`.

Si vous ne désirez pas créer d'utilisateur, vous pouvez utiliser en local l'utilisateur `root`, configuré par défaut lors de l'installation de la base MySQL. Dans ce cas, il vous suffit d'indiquer `root` comme login et de ne pas préciser de mot de passe (Attention ! pour des raisons de sécurité évidentes, cette configuration est strictement réservée à un usage local).

### Création d'un nouveau site dans Dreamweaver

Avant de créer le fichier de connexion à la base, il faut configurer un nouveau site correspondant au projet Sport (revoir si besoin la démarche, présentée dans le chapitre 4). L'étape 5 de la procédure décrite ci-dessous, dédiée à la configuration d'un serveur d'évaluation, permettra par la suite de créer automatiquement le fichier de connexion à la base de données. Si vous désirez configurer vous-même ce fichier en mode code, validez votre nouveau site à partir de l'étape 4 :

1. Depuis le menu de l'interface de Dreamweaver, cliquez sur la rubrique Sites puis sélectionnez Gérer les sites.
2. Dans la boîte de dialogue Gérer les sites, cliquez sur le bouton Nouveau.
3. Dans la boîte de dialogue Définition du site, cliquez sur l'onglet Avancé s'il n'est pas déjà sélectionné.

4. Cliquez sur la catégorie Infos Locales (voir figure 18-2) et renseignez les champs suivants :
  - Nom du site : SITEsport ;
  - Dossier racine : cliquez sur le petit dossier situé à droite du champ afin d’ouvrir l’explorateur de fichier. Créez puis sélectionnez un nouveau dossier /SITEsport/ dans le répertoire racine www de la suite Wamp 5.
5. Cliquez ensuite sur la catégorie Serveur d’évaluation (voir repère 1 de la figure 18-3) et renseignez les champs suivants (si vous ne désirez pas utiliser le mode assisté de Dreamweaver pour créer votre fichier de connexion, cette étape 5 n’est pas nécessaire) :
  - Modèle de serveur : sélectionnez PHP-MySQL dans le menu déroulant (voir repère 2 de la figure 18-3) ;
  - Accès : sélectionnez Local/Réseau dans le menu déroulant (voir repère 3 de la figure 18-3) ;
  - Dossier du serveur d’évaluation : théoriquement, ce champ doit être préconfiguré avec le chemin menant au répertoire du site www/SITEsport (voir repère 4 de la figure 18-3) ;
  - Préfixe de l’URL : ajoutez SITEsport/ à la suite de http://localhost/ (voir repère de la 5 figure 18-3).
6. Cliquez sur le bouton OK de la boîte de dialogue puis sur le bouton Terminer de la boîte Gérer les sites.

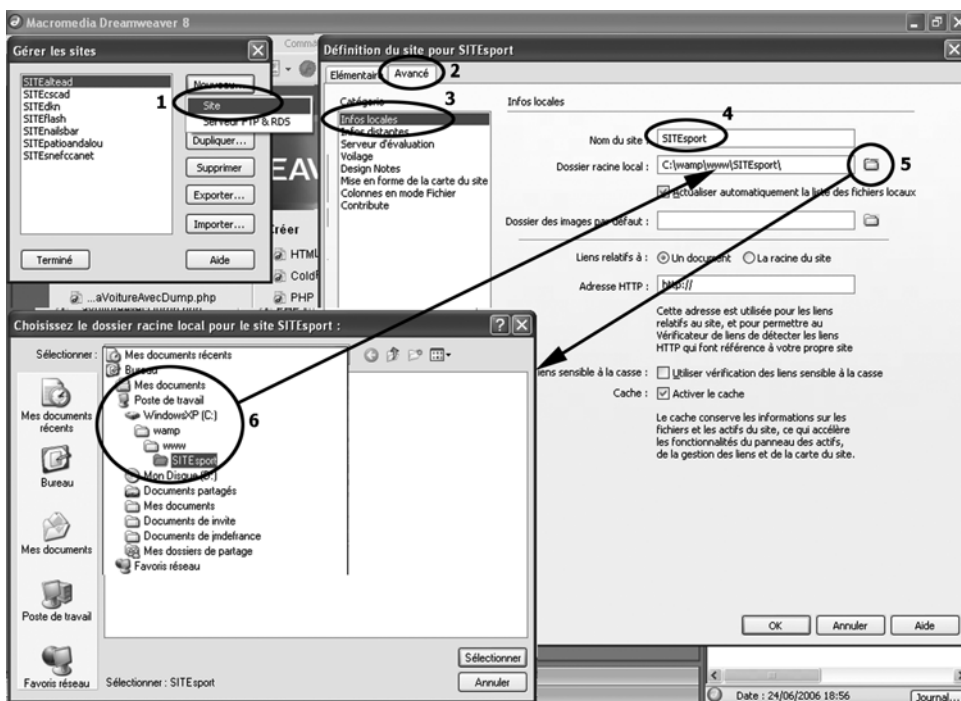
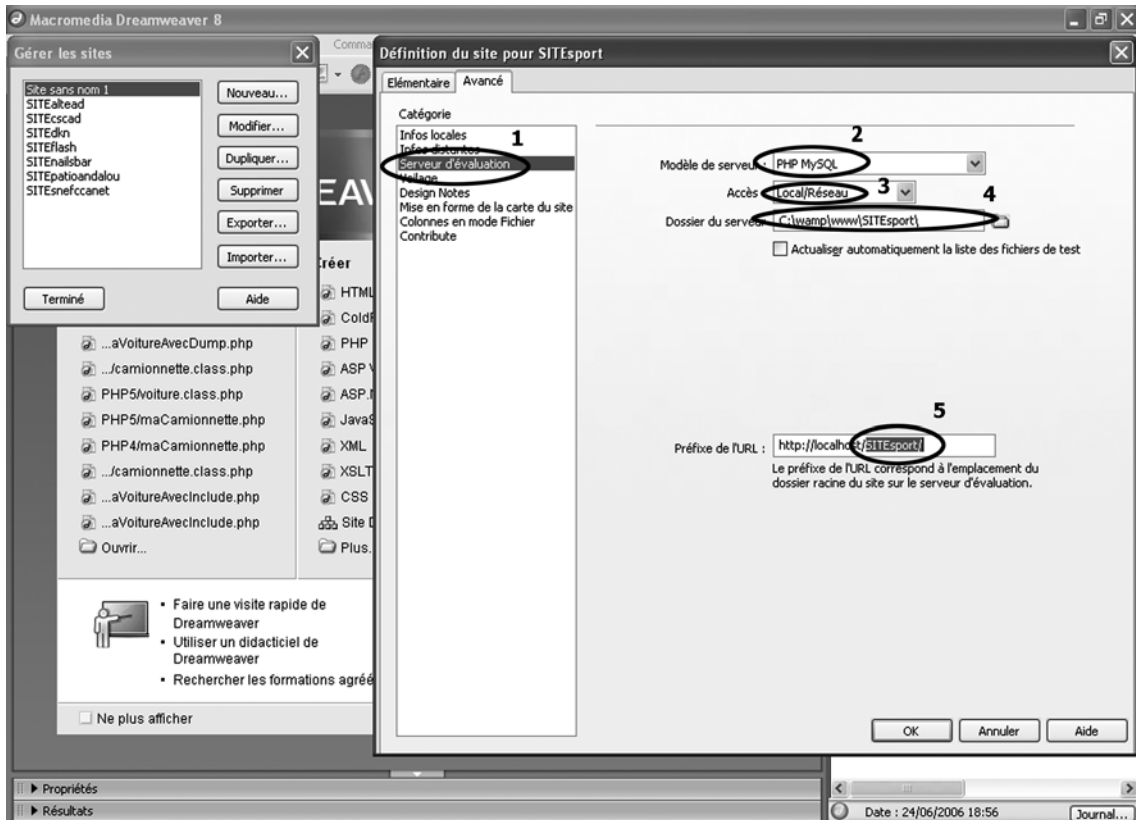


Figure 18-2

Configuration de la catégorie Infos locales d’un nouveau site destiné à tester l’application dynamique Sport

**Figure 18-3**

Configuration de la catégorie *Serveur d'évaluation*. La configuration de cette catégorie est nécessaire si l'on désire créer un fichier de connexion conforme aux comportements serveur de Dreamweaver.

### Création avec la boîte de dialogue de Dreamweaver

Voici la procédure à suivre pour créer un fichier de connexion à la base MySQL à l'aide de l'assistant de Dreamweaver. Avant de commencer, assurez-vous que Wamp 5 est démarré.

1. Ouvrez une page PHP dans Dreamweaver (menu **Fichier**>**Nouveau**, sélectionnez **Page dynamique** et **PHP** puis cliquez sur **Créer**) (voir figure 18-4).
2. Déroulez le panneau **Application** et sélectionnez l'onglet **Base de données** (voir figure 18-5).
3. Cliquez sur le bouton **+** et sélectionnez la rubrique **Connexion MySQL** qui s'affiche dans le menu contextuel (voir figure 18-5).

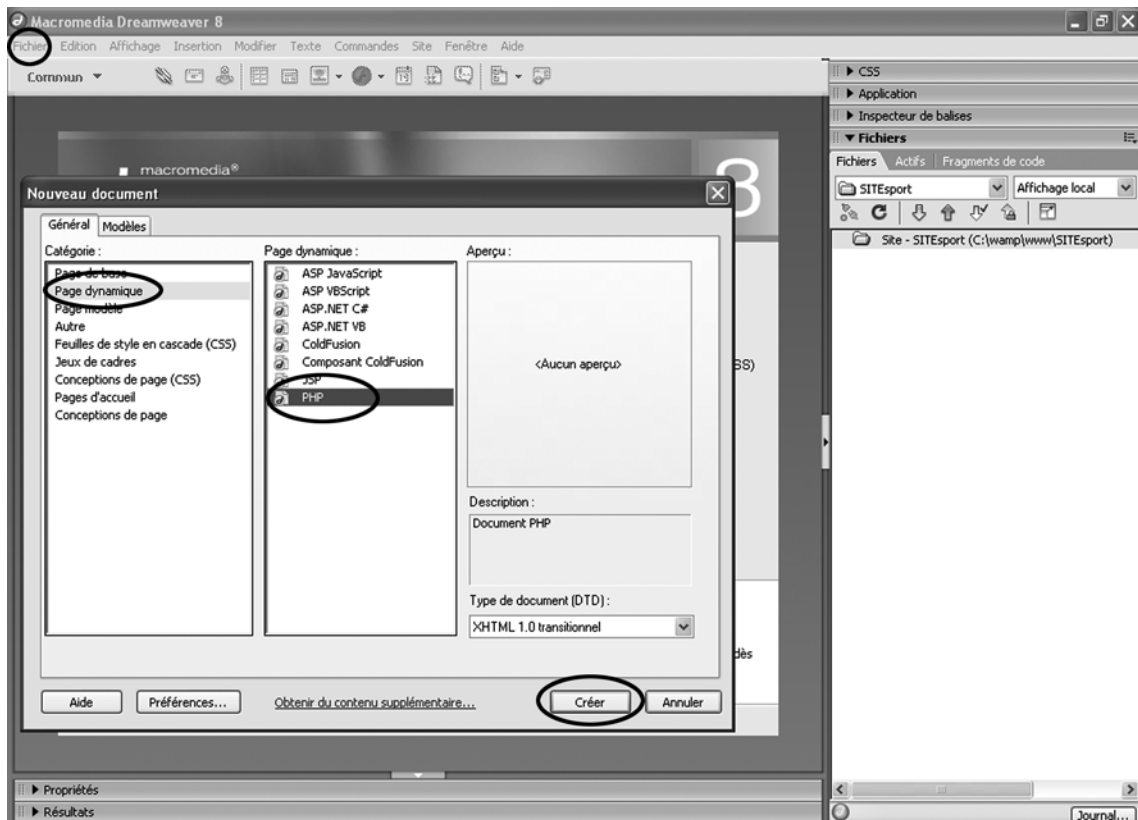


Figure 18-4

Pour créer une connexion, commencez par ouvrir une page dynamique PHP.

4. La boîte de dialogue Connexion MySQL s'affiche (voir figure 18-5).
5. Saisissez un nom pour la connexion que vous allez créer. Ce nom doit être explicite et ne doit pas comporter d'espace. Dans le cadre de notre application, nous utiliserons ConnexionSport.
6. Dans le champ Serveur MySQL, indiquez l'ordinateur sur lequel est installé le serveur MySQL. Cela peut être une adresse IP ou un nom de serveur ; dans notre cas, le serveur MySQL étant installé sur le même serveur que PHP, nous saisissons localhost.
7. Saisissez ensuite les paramètres de l'utilisateur sport : nom de l'utilisateur = sport et mot de passe = eyroll es. Si vous n'avez pas défini l'utilisateur sport dans la base MySQL, vous pouvez saisir provisoirement l'utilisateur installé par défaut sur la base MySQL : nom de l'utilisateur = root, sans mot de passe.

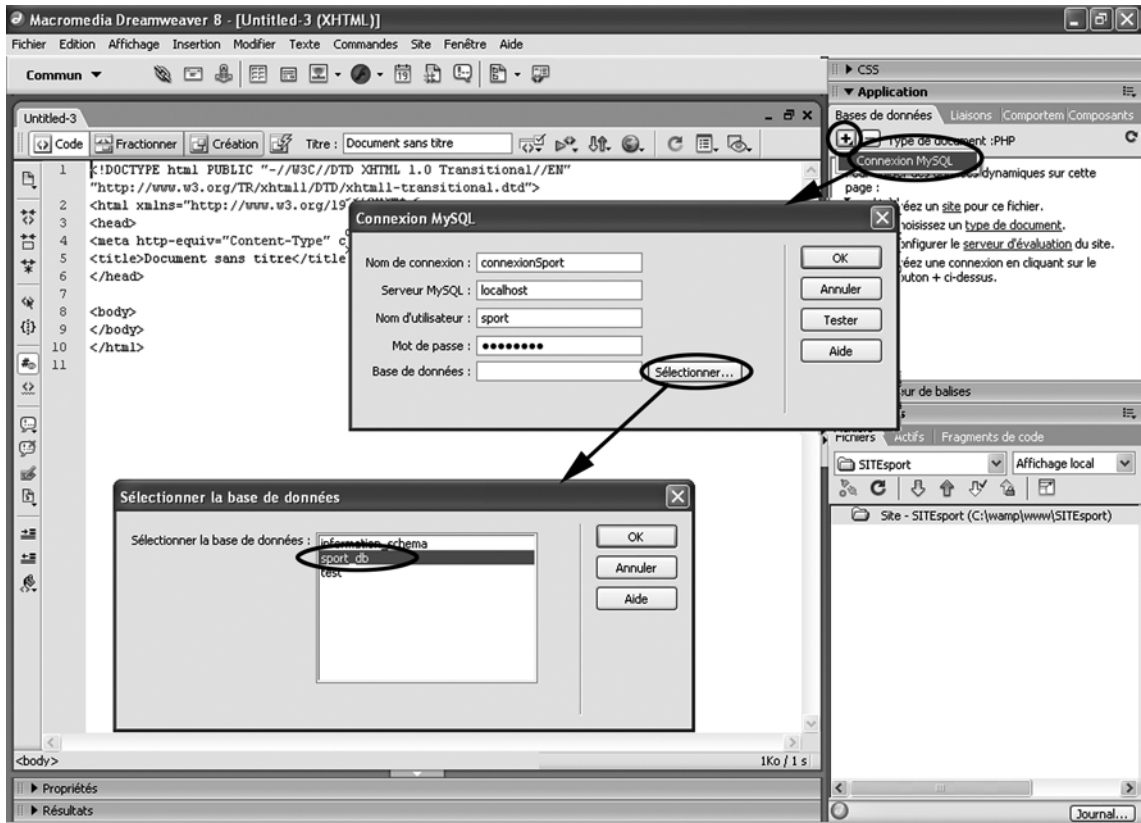


Figure 18-5

Depuis le panneau *Base de données*, cliquez sur **+** et sélectionnez *Connexion MySQL* pour ouvrir la fenêtre de paramétrage de la connexion à la base. Saisissez les paramètres du compte utilisateur *Sport* dans la boîte de dialogue puis cliquez sur le bouton *Sélectionner* pour afficher la liste des bases disponibles sur le serveur MySQL.

8. Cliquez sur le bouton *Sélectionner* pour afficher toutes les bases de données disponibles (voir figure 18-5). Sélectionnez la base *sport\_db* et validez en cliquant sur le bouton *OK*.
9. La base sélectionnée s'affiche dans le champ *Base de données*. Vérifiez que la connexion est valide en cliquant sur le bouton *Tester*. Un message vous informe que la connexion est établie si tous vos paramètres sont corrects (voir figure 18-6). Si la connexion échoue, vérifiez de nouveau les paramètres ci-dessus (Serveur, Nom, Mot de passe) et assurez-vous que Wamp 5 est actif.

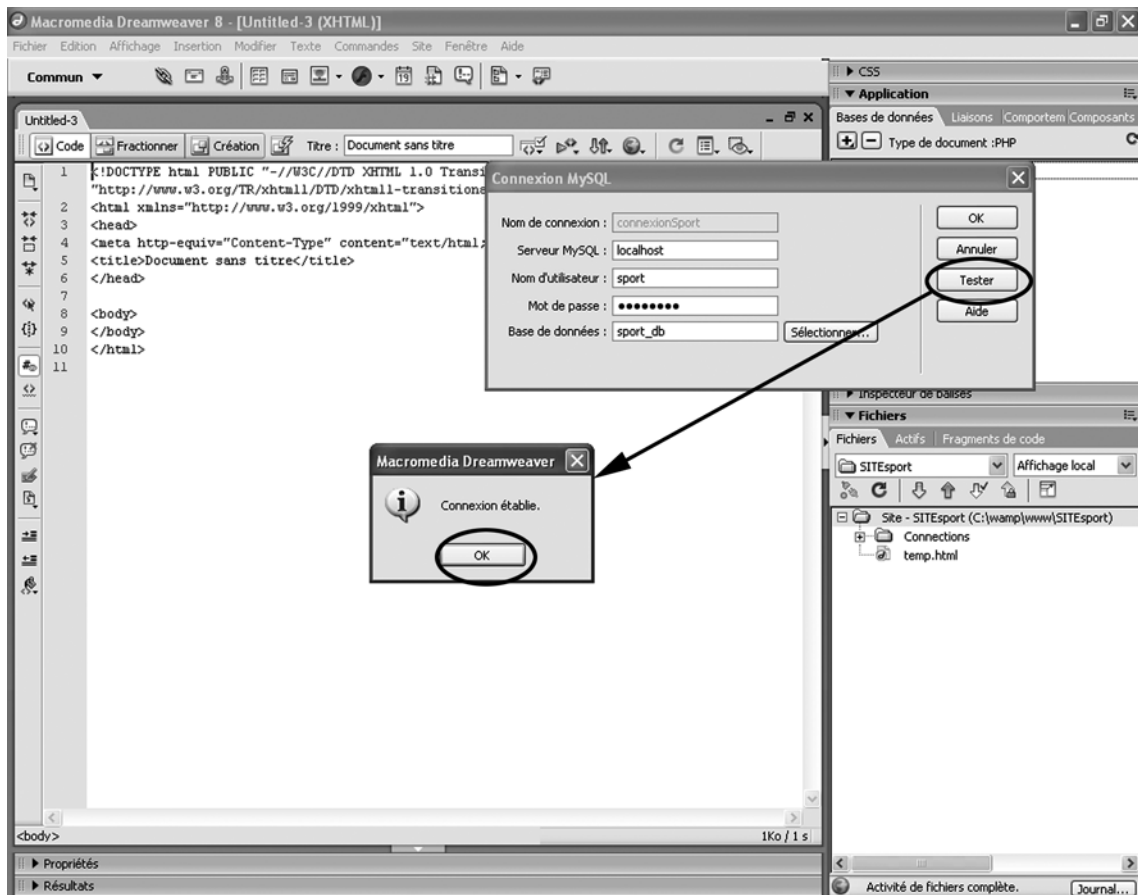


Figure 18-6

Avant de valider votre connexion, vous pouvez tester si tous les paramètres sont corrects en cliquant sur le bouton *Tester*.

10. Fermez la fenêtre du message et confirmez la création de la connexion en cliquant sur le bouton **OK**.
11. La connexion à la base de données est désormais établie. Une icône représentant la base `sport_db` apparaît dans la fenêtre Base de données. Vous pouvez cliquer successivement sur les + qui précèdent les noms des branches de l'arborescence (`ConnexionSport`, puis `Table`, puis `adherents`, par exemple) pour déplier l'arbre de la base de données et faire apparaître les différents champs qui constituent chacune de ses tables (voir figure 18-7).

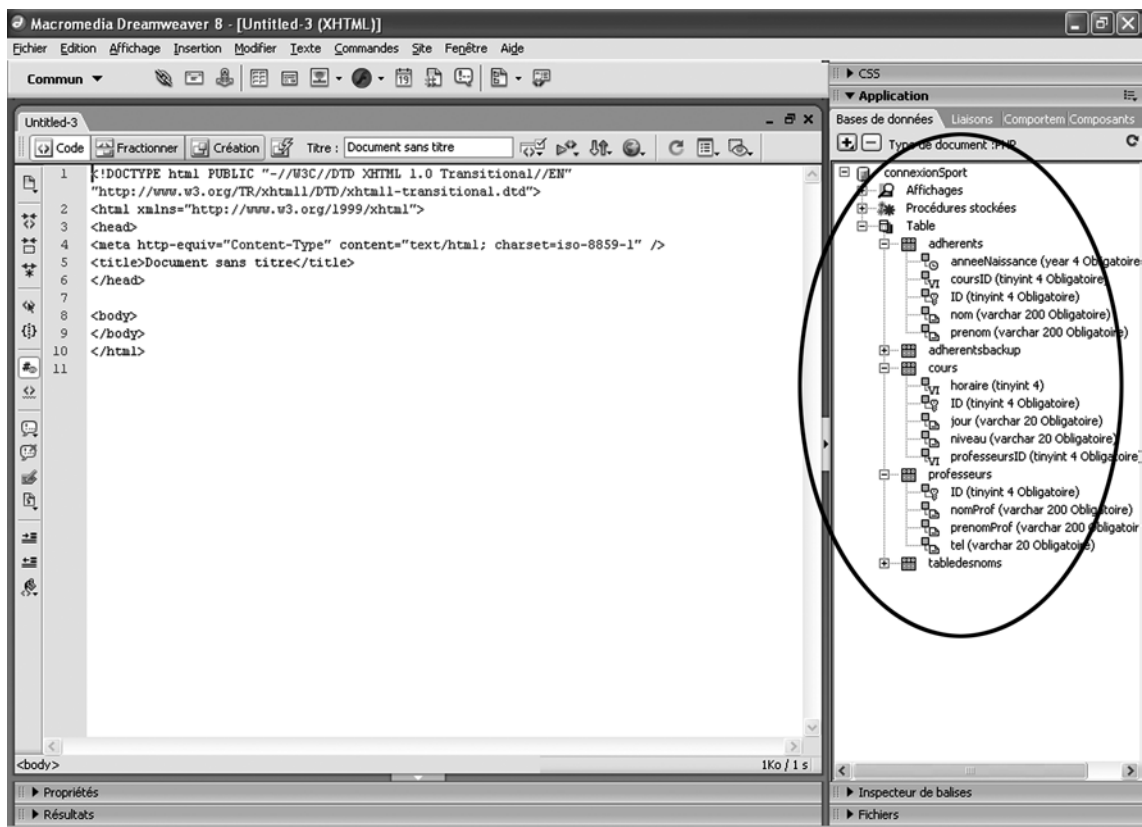


Figure 18-7

Une fois la connexion établie, vous pouvez consulter tous les champs disponibles dans chacune des tables de la base de données `sport_db`.

### Attention ! Le répertoire `Connections` ne doit pas être supprimé !

Après la création d'une connexion, un sous-répertoire est automatiquement créé dans l'arborescence du site (`C : wamp/www/SITESport/`). Il se nomme `Connections` et contient un fichier PHP qui porte le nom de la connexion qui vient d'être configurée (voir figure 18-8). Ce fichier regroupe tous les paramètres de votre connexion (nom du serveur MySQL, login du compte, mot de passe, nom de la base de données) et doit être transféré sur le serveur distant lors de la publication du site. Évidemment, ce répertoire et ce fichier ne doivent en aucun cas être supprimés, au risque d'interrompre toutes les interactions entre la base de données et les scripts dynamiques du serveur.

12. Lors de la création d'une connexion, un répertoire `/Connections/` et un fichier `connexionSport.php` sont créés automatiquement. Si vous ouvrez le fichier `connexionSport.php` dans l'éditeur de code de Dreamweaver, vous remarquerez qu'il contient tous les paramètres de la connexion qui vient d'être créée (voir figure 18-8). Pour ouvrir `connexionScore.php`, double-cliquez sur le nom du fichier dans le panneau Fichier, puis cliquez sur le bouton Afficher en mode code si nécessaire.

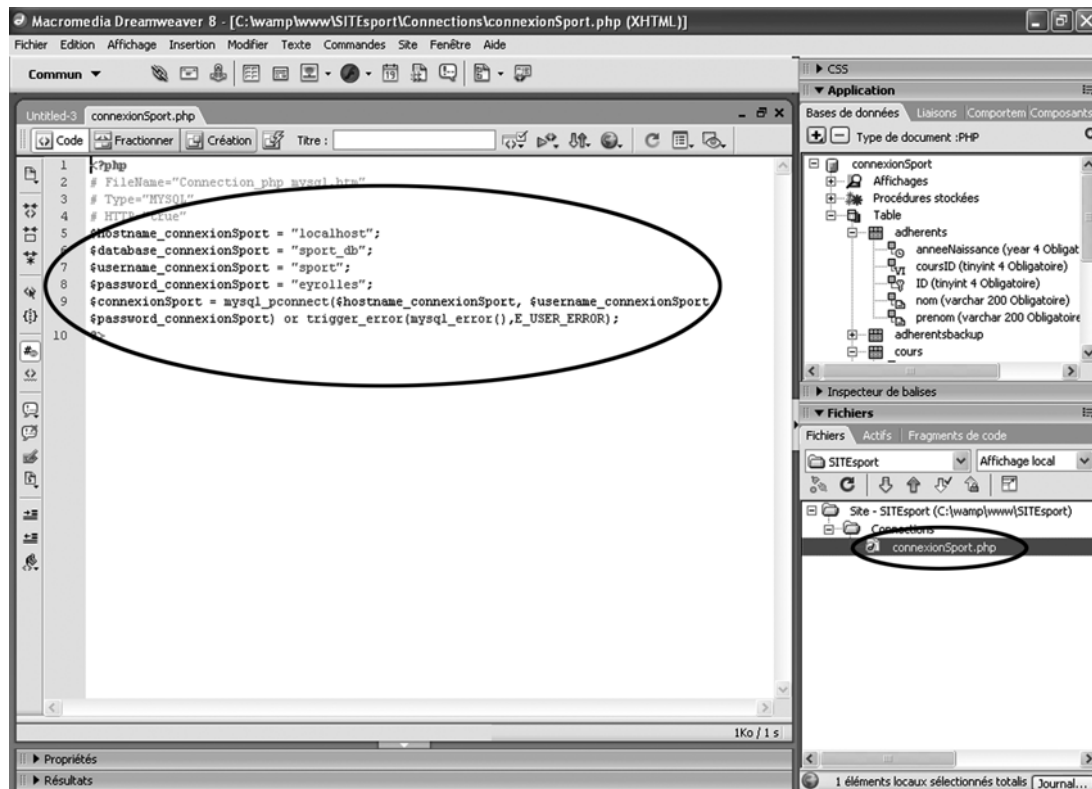


Figure 18-8

Lors de la création d'une connexion, un répertoire *Connexions* et un fichier portant le nom de la connexion *ConnexionSport.php* sont automatiquement créés dans l'arborescence du site.

### Création en mode code à l'aide de l'éditeur

L'autre solution pour disposer d'un fichier de connexion consiste à saisir les paramètres de connexion et les fonctions en mode Code dans l'éditeur de votre choix. Afin que les scripts de cet ouvrage puissent être exploités de la même manière quel que soit le mode de création choisi, nous utiliserons la même convention de nommage des variables que celle utilisée dans le fichier `connexionScore.php` généré automatiquement par Dreamweaver dans le mode de création précédent (voir le code ci-dessous) :

```

<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_connexionSport = "localhost";
$database_connexionSport = "sport_db";
$username_connexionSport = "sport";
$password_connexionSport = "eyrolles";
$connexionSport = mysql_pconnect($hostname_connexionSport, $username_connexionSport,
➤$password_connexionSport) or trigger_error(mysql_error(),E_USER_ERROR);
?>

```

Pour créer votre fichier de connexion en mode code, saisissez les instructions ci-dessus dans votre éditeur et enregistrez-les dans un fichier PHP nommé `connexionSport.php` dans un répertoire `/Connections/`.

### Descriptif du fichier de connexion

Les trois premières lignes sont des commentaires (leur saisie n'est donc pas indispensable).

Les quatre lignes suivantes correspondent à l'initialisation des variables de connexion. Ainsi, le nom de la machine hôte (`hostname`) est affecté par la valeur `localhost` car le serveur MySQL se trouve sur la même machine que le serveur Web ; le nom de la base (`database`) est affecté par `sport_db`, qui est le nom de la base à laquelle vous désirez vous connecter ; le nom d'utilisateur et le mot de passe (`username` et `password`) permettent d'être authentifié par la base comme un utilisateur valide (pour ces deux derniers paramètres, vous pouvez utiliser les valeurs `sport` et `eyrolles` ou `root` sans mot de passe si l'utilisateur `sport` n'a pas été créé).

La ligne qui suit ces initialisations exploite la fonction `mysql_pconnect()` et reprend en paramètres trois des variables de connexion précédentes. Cette fonction envoie une demande de connexion au serveur MySQL. Celui-ci, après avoir vérifié que l'utilisateur et son mot de passe sont valides, retourne un identifiant de connexion mémorisé dans la variable `$connexionSport`. Si une des informations est invalide, la fonction retourne la valeur `FALSE` et un message d'erreur (`trigger_error()`) indique que la connexion a échoué.

#### À noter

Si vous désirez gérer vous-même les problèmes de connexion (en faisant un simple test sur l'identificateur retourné par la fonction, par exemple), ajoutez le caractère `@` devant la fonction afin d'éviter qu'elle ne génère automatiquement un message d'erreur à l'écran (exemple : `@mysql_connect()`).

Il existe deux déclinaisons de cette fonction PHP de connexion à un serveur MySQL :

- `mysql_connect()` : connexion éphémère au serveur. Cette connexion se termine par l'appel de la fonction `mysql_close()` ou à la fin du script ;
- `mysql_pconnect()` : connexion persistante au serveur. Contrairement à la connexion éphémère, elle ne se termine pas par l'appel de la fonction `mysql_close()` ou à la fin du script mais est conservée pour un prochain accès, d'où son nom. Ce type de connexion est intéressant pour des applications très sollicitées (les moteurs de recherche, par exemple). Cependant, MySQL ayant un processus d'établissement de connexions relativement efficace, en règle générale, il est souvent préférable d'utiliser des connexions éphémères.

#### Attention aux connexions persistantes !

Si vous utilisez la fonction de connexion persistante `mysql_pconnect()` (et non `mysql_connect()`) et que la configuration du serveur Web Apache ou de MySQL n'est pas adaptée, vous risquez d'atteindre rapidement le nombre maximal de connexions et de bloquer votre serveur MySQL (apparition du message « Too Many Connections »). Dans ce cas, si vous ne pouvez pas modifier la configuration de vos serveurs, remplacez votre fonction de connexion persistante par une fonction de connexion éphémère (`mysql_connect()`).

## Concept du jeu d'enregistrements

Une fois que le « canal » d'accès au serveur MySQL est mis en place, vous pouvez exploiter les enregistrements de la base et extraire un sous-ensemble de données d'une ou plusieurs tables. Ce sous-ensemble de données s'appelle un jeu d'enregistrements et forme une nouvelle table puisqu'il est formé de champs et d'enregistrements selon la requête d'extraction employée.

Pour créer un jeu d'enregistrements, vous allez devoir élaborer une requête SQL caractérisée par les critères de sélection des enregistrements et par les champs demandés (revoir si besoin le chapitre 17 consacré aux commandes SQL).

Nous rappelons que pour transmettre une requête SQL à la base de données, il est indispensable qu'une connexion soit préalablement établie avec le serveur MySQL. En effet, la requête et le jeu d'enregistrements emprunteront le « canal » de la connexion `connexionSport`. Les différentes informations qui constituent une requête permettent de sélectionner la table (ou les tables) de la base à laquelle la connexion donne accès et de définir les critères de sélection des enregistrements et les noms des champs qu'on désire exploiter dans la page dynamique du site (voir le repère 1 de la figure 18-9). En réponse, le serveur MySQL renvoie un jeu d'enregistrements (voir le repère 2 de la figure 18-9). Ces enregistrements sont ensuite utilisés comme éléments dynamiques dans la page PHP qui a généré la requête.

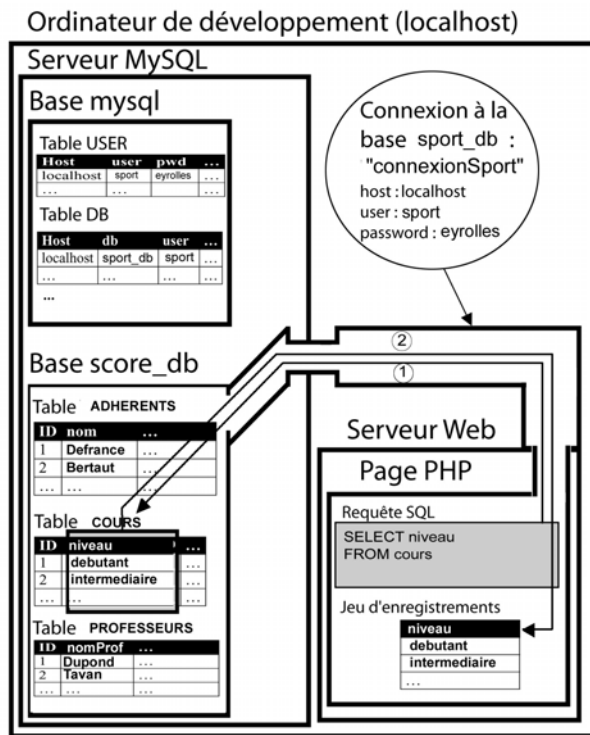


Figure 18-9

*Schéma de principe d'une requête et d'un jeu d'enregistrements : la requête SQL et le jeu d'enregistrements utilisent le « canal » de la connexion à la base de données pour passer du serveur Web au serveur MySQL et vice versa.*

## Création d'un jeu d'enregistrements

### Bibliothèque de fonctions MySQL

Nous disposons maintenant d'un fichier (dans notre exemple, le fichier `connexionScore.php`) regroupant les paramètres de connexion ainsi que la fonction de connexion (`mysql_connect()`, qui retourne l'identificateur de connexion nommé `$connexionSport` dans notre exemple).

Nous pouvons désormais construire une première page dynamique qui affichera les informations issues de la base de données. Nous allons envoyer une requête SQL à la base de données afin de récupérer les différents niveaux de la table `cours`. Pour cela, nous avons besoin d'autres fonctions PHP dédiées à la gestion de MySQL (revoir le tableau 6-18 ou voir le tableau récapitulatif 18-1). Ces fonctions PHP-MySQL sont regroupées dans une bibliothèque (ou extension) et permettent, entre autres, de se connecter au serveur MySQL, d'exécuter des requêtes SQL ou encore de récupérer les résultats en les mettant en forme pour les afficher dans la page dynamique.

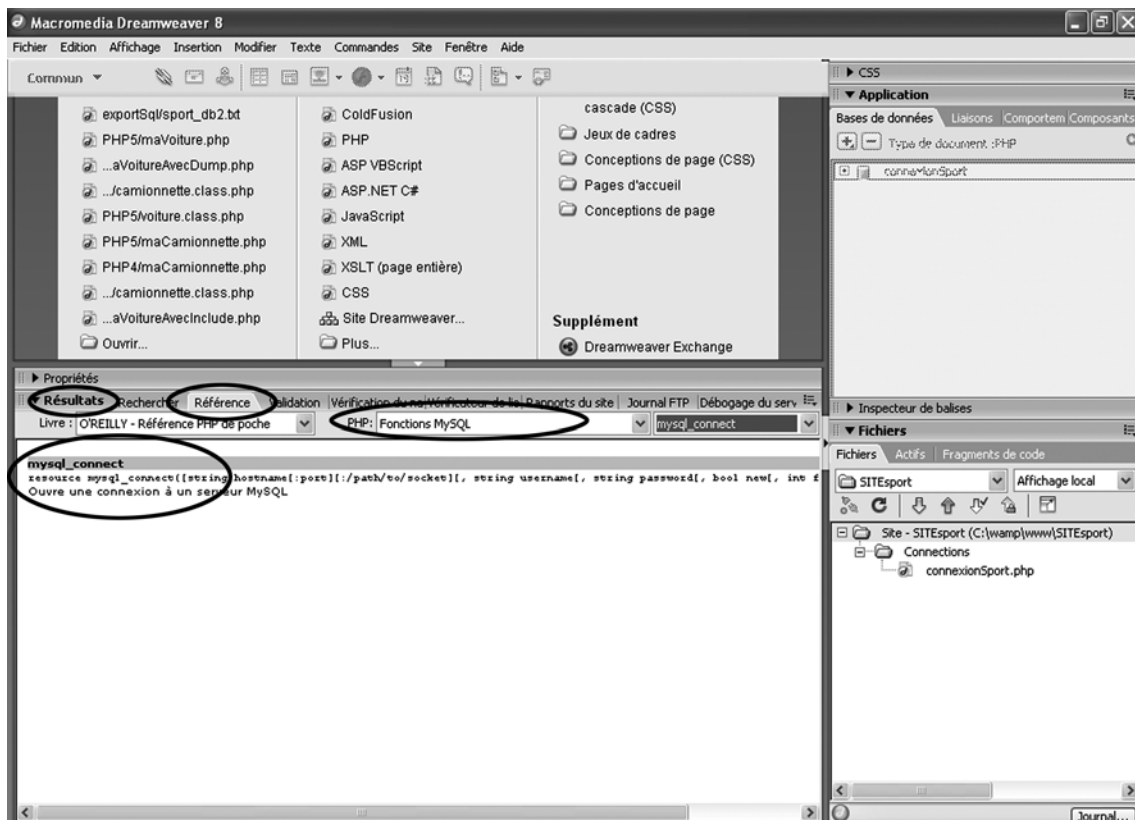


Figure 18-10

L'onglet *Référence* du panneau *Résultats* de Dreamweaver permet d'accéder à la syntaxe et à la description de chacune des fonctions PHP dédiées à MySQL.

**À noter**

Ces fonctions sont facilement identifiables car leur nom commence toujours par le préfixe `mysql_`. Vous pouvez d'ailleurs consulter la liste exhaustive de ces fonctions MySQL depuis Dreamweaver à partir du panneau Résultats, onglet Référence (sélectionnez ensuite dans les menus déroulants Livre : Référence PHP de poche : puis Fonctions MySQL ; voir figure 18-10). Notez que ce même onglet Référence permet aussi d'accéder aux différentes commandes SQL, présentées au chapitre 17 de cet ouvrage (voir figure 18-11).

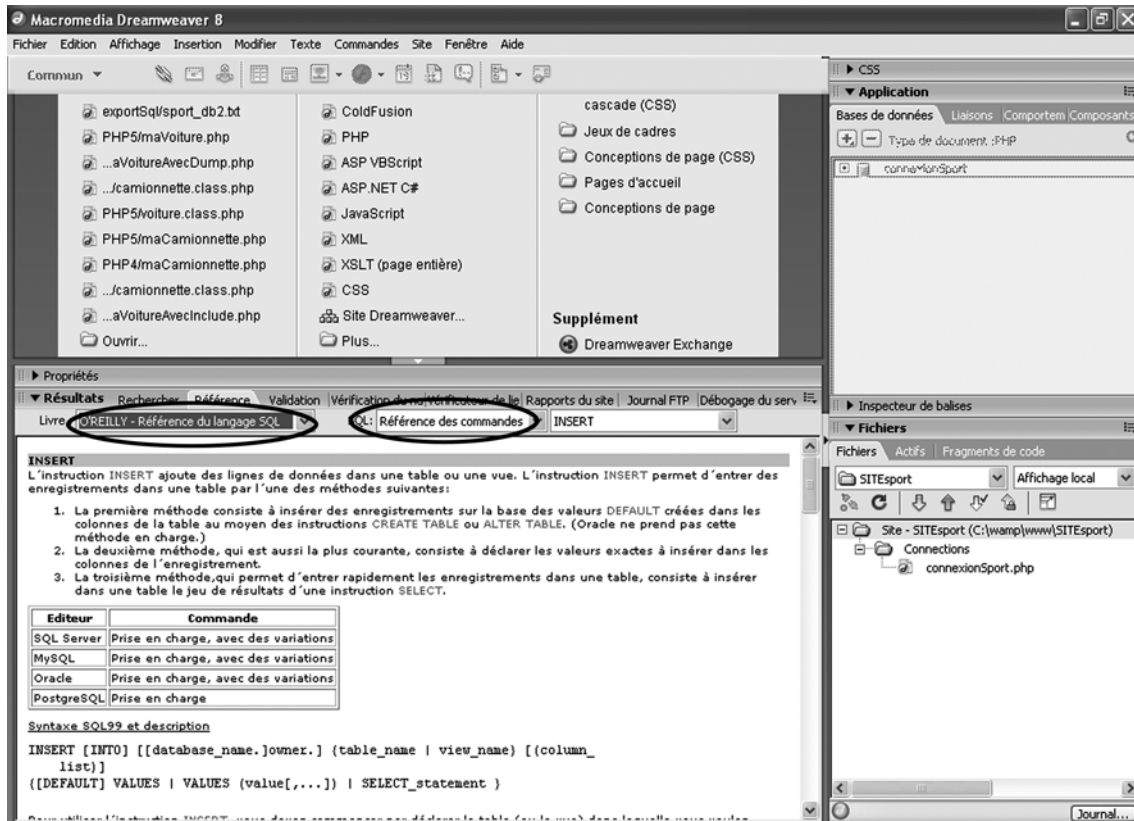


Figure 18-11

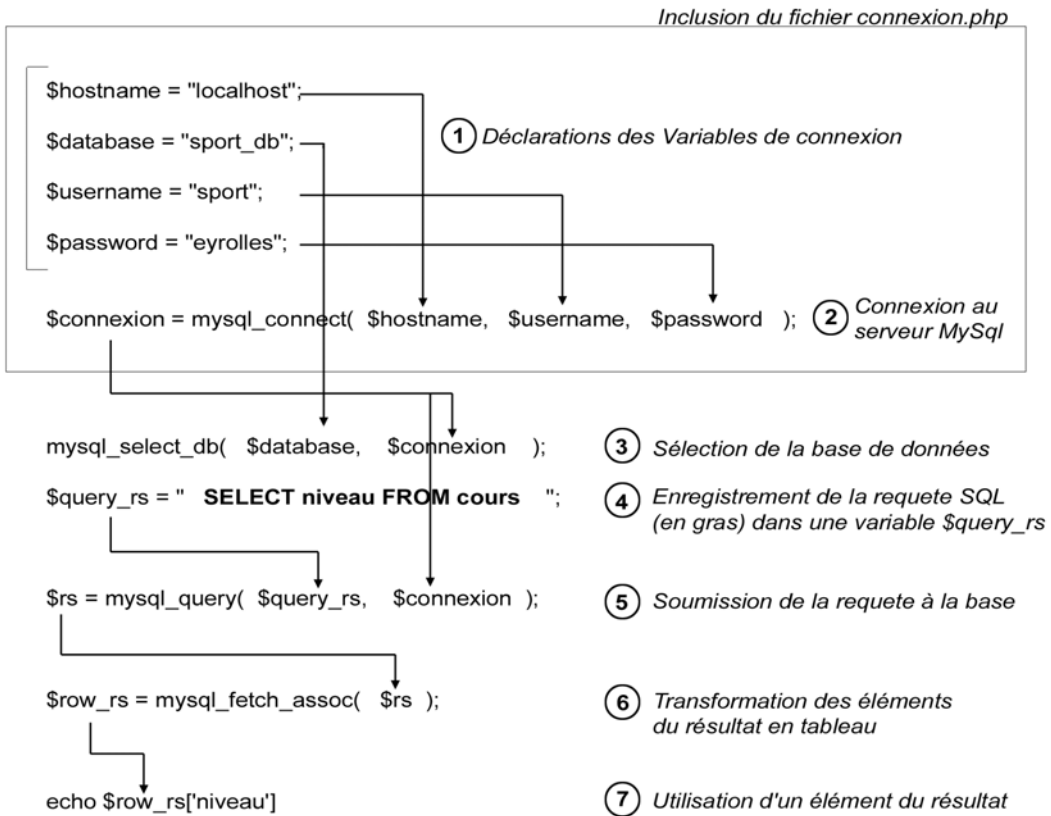
L'onglet Référence du panneau Résultats de Dreamweaver permet aussi d'accéder à la syntaxe des commandes SQL de MySQL.

**Tableau 18-1. Tableau récapitulatif des principales fonctions PHP dédiées à MySQL**

Fonction PHP	Description
mysql_connect()  mysql_pconnect()	<p>Permet d'établir une connexion (ou encore un « canal » ; voir figure 18-1) avec le serveur MySQL selon les paramètres de l'utilisateur (nom et mot de passe) et d'un serveur spécifique (exemple : monserveur.com ou localhost si le serveur MySQL est installé sur la même machine que le serveur Web).</p> <p>Cette fonction renvoie un identificateur qui est ensuite utilisé pour dialoguer avec le serveur. Cet identificateur peut être mémorisé dans une variable (par exemple : \$connexion) afin de faciliter son intégration dans les autres fonctions PHP de la bibliothèque MySQL (mysql_select_db() ou mysql_query() par exemple). Si la connexion avec le serveur MySQL est impossible, l'identificateur renvoyé est égal à 0 et un message d'erreur s'affiche à l'écran. Le fait que la valeur retournée en cas d'échec soit nulle permet de s'assurer facilement que la connexion est bien établie à l'aide d'un simple test : <code>if(!\$connexion) { //suite des fonctions }</code>.</p> <p>À noter : Une déclinaison de cette fonction (mysql_pconnect()) permet d'obtenir une connexion persistante à la base.</p>
mysql_select_db()	<p>Permet de sélectionner une base de données du serveur MySQL pour une connexion spécifique. Par exemple, la fonction mysql_select_db("sport_db", \$connexion) permet de présélectionner la base sport_db pour toutes les requêtes utilisant l'identifiant de connexion \$connexion.</p>
mysql_query()	<p>Permet d'exécuter une requête SQL (SELECT) mais aussi tout autre type de commande MySQL (INSERT, UPDATE, etc.).</p> <p>Cette fonction renvoie un pointeur qui permet d'exploiter les résultats de la requête (appelé aussi « jeu d'enregistrements » ou « RecorSet » : rs).</p> <p>En général, la valeur de ce pointeur est affectée à une variable du script (par exemple : \$rs=mysql_query(\$query,\$connexion)).</p> <p>À noter : Ce pointeur ne peut pas être directement utilisé ; il faut donc faire appel à une autre fonction (par exemple mysql_fetch_row() ou mysql_fetch_asso()) pour récupérer les différentes lignes du résultat.</p>
mysql_fetch_row()	<p>Permet de récupérer une des lignes du jeu d'enregistrements et positionne le pointeur initial sur la ligne suivante. Avec cette fonction, chaque ligne de résultat est renvoyée sous forme d'un tableau indicé. En général, une variable de type tableau est alors affectée par le résultat retourné afin de pouvoir facilement l'exploiter au sein du script. Par exemple, si \$row=mysql_fetch_row(\$rs), on peut ensuite afficher la valeur de premier champ avec l'instruction suivante : <code>echo \$row[0]</code> ;</p>
mysql_fetch_asso()	<p>Permet de récupérer une des lignes du jeu d'enregistrements et positionne le pointeur initial sur la ligne suivante. Avec cette fonction, chaque ligne de résultat est renvoyée sous forme d'un tableau associatif. En général, une variable de type tableau est alors affectée par le résultat retourné afin de pouvoir facilement l'exploiter au sein du script. Par exemple, si \$row=mysql_fetch_asso(\$rs), on peut ensuite afficher la valeur du champ nom avec l'instruction suivante : <code>echo \$row['nom']</code> ;</p>
mysql_fetch_array()	<p>Permet de récupérer une des lignes du jeu d'enregistrements et positionne le pointeur initial sur la ligne suivante. Avec cette fonction, chaque ligne de résultat peut être renvoyée sous forme d'un tableau indicé, d'un tableau associatif ou d'un tableau mixte selon un argument optionnel de la fonction</p>
mysql_fetch_object()	<p>Permet de récupérer une des lignes du jeu d'enregistrements et positionne le pointeur initial sur la ligne suivante. Avec cette fonction, chaque ligne de résultat peut être renvoyée sous forme d'un objet. Par exemple, si \$row=mysql_fetch_object(\$rs), on peut ensuite afficher la valeur du champ nom avec l'instruction suivante : <code>echo \$row-&gt;nom</code> ;</p>
mysql_free_result()	<p>Permet de libérer la mémoire utilisée par un jeu d'enregistrements. Cette fonction est particulièrement recommandée lorsque des jeux d'enregistrements risquent de solliciter beaucoup de mémoire au sein d'un même script. À noter : Si cette fonction n'est pas utilisée, la mémoire allouée aux jeux d'enregistrements sera automatiquement vidée à la fin du script.</p>

## Procédure de création d'un jeu d'enregistrements

Pour créer un jeu d'enregistrements dans une nouvelle page dynamique, il faut commencer par établir une connexion avec le serveur MySQL. Pour cela, il suffit d'appeler (à l'aide de la fonction `require_once()`) le fichier de connexion.



**Figure 18-12**

Détail d'un script d'envoi d'une requête SQL et de la mise en forme du jeu d'enregistrements correspondant

Lors de l'exécution de la page, tout le contenu du fichier est copié dans votre page dynamique à la place de l'instruction `require_once()` (voir repères 1 et 2 de la figure 18-12). Vous pourrez ensuite faire référence aux variables `$database` et `$connexion` dans les différentes fonctions de la page (voir repères 3 et 5 de la figure 18-12).

Une fois la connexion établie avec le serveur MySQL, sélectionnez la base de données sur laquelle vous désirez intervenir à l'aide de la fonction `mysql_select_db()` (voir repère 3 de la figure 18-12). Le nom de la base à sélectionner et la connexion concernée sont transmis dans les arguments de la fonction.

L'étape suivante consiste à élaborer la requête SQL qui sera transmise à la base précédemment sélectionnée. Pour cela, vous pouvez enregistrer cette requête dans une variable spécifique nommée `$query_rs` (voir repère 4 de la figure 18-12). Cette variable est ensuite transmise en argument dans la fonction de soumission `mysql_query()` (voir repère 5 de la figure 18-12). L'autre argument de la fonction permet de préciser que la requête doit être transmise à la base de donnée par l'intermédiaire de la connexion préalablement établie (soit `$connexion` dans notre exemple). En réponse à la requête, la fonction retourne un pointeur qui est enregistré dans la variable `$rs` (`rs` pour RecordSet : jeu d'enregistrements).

Le pointeur `$rs` ne peut pas être exploité directement. Il faut pour ce faire utiliser une fonction de mise en forme du jeu d'enregistrements (voir repère 6 de la figure 18-12). Selon les besoins de l'application, plusieurs types de fonctions peuvent être retenues pour effectuer cette transformation : (`mysql_fetch_row()`, `mysql_fetch_asso()`, `mysql_fetch_array()` ou encore `mysql_fetch_objet()`). À chaque appel de ces fonctions, une ligne du jeu d'enregistrements devient accessible sous forme d'un tableau de variables (voir repère 7 de la figure 18-12) et le pointeur se déplace automatiquement sur la ligne suivante. Lorsque le jeu d'enregistrements est complètement parcouru, le pointeur retourne une valeur `FALSE`. Cela permet d'intégrer facilement la gestion de ces fonctions dans une structure de boucle (`while` ou `for`).

### Exemple de création d'un jeu d'enregistrements

Vous allez maintenant passer à la pratique et créer le script d'une page dynamique :

1. Dans Dreamweaver, ouvrez un nouveau document PHP (menu Fichier>Nouveau>Page dynamique>PHP). Enregistrez-le sous le nom `requete1.php` dans le répertoire `/sport` (répertoire à créer si nécessaire). Il est préférable d'enregistrer le fichier tout de suite afin de pouvoir déterminer sans ambiguïté le chemin relatif d'accès au fichier `connexionSport.php` que nous allons créer dans l'étape suivante.
2. Afin de disposer des paramètres du fichier de connexion précédemment créé, vous allez ajouter une commande `require_once()` au début du fichier (voir repère 1 de la figure 18-13). N'oubliez pas de saisir au préalable la balise d'ouverture de script PHP : `<?php`. Lors de l'interprétation du script par le module PHP, cette ligne sera automatiquement remplacée par le contenu du fichier `connexionSport.php` (revoir figure 18-12).

```
require_once('../Connections/connexionSport.php');
```

3. À la ligne suivante, ajoutez la fonction `mysql_select_db()` en précisant en argument les variables `$database_connexionSport` et `$connexionSport` afin de sélectionner la base `sport_db` pour les prochaines utilisations de la connexion précédemment créée (`$connexionSport`).

```
mysql_select_db($database_connexionSport, $connexionSport);
```

4. Saisissez ensuite la requête SQL `SELECT niveau FROM cours` et enregistrez-la dans une variable `$query_rsRappelNiveaux`.

```
$query_rsRappelNiveaux = "SELECT niveau FROM cours";
```

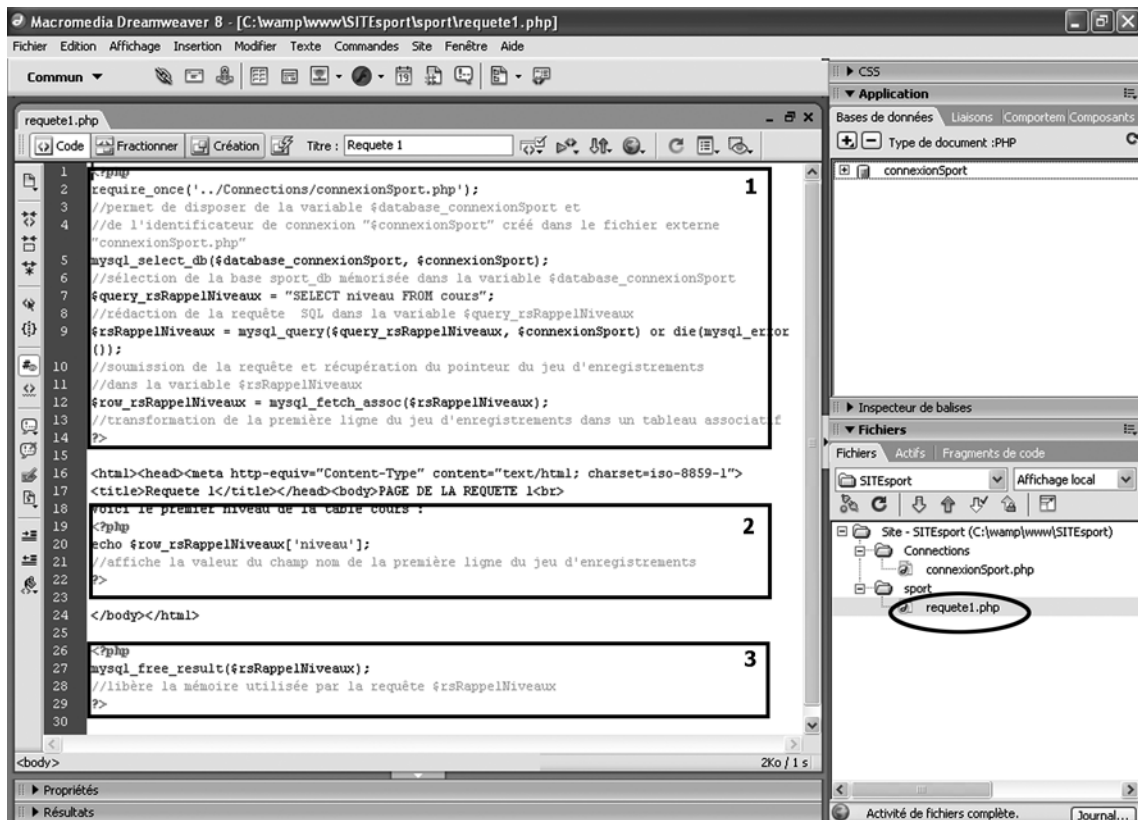


Figure 18-13

Script de la page dynamique *requete1.php*

- Configurez la fonction de soumission de la requête en indiquant en argument le nom de la variable précédemment créée (contenant la requête SQL) et le nom de la variable de l'identificateur de la connexion (`$connexionSport`). Vous pouvez compléter cette instruction par une fonction qui affichera le détail de l'erreur si la soumission est impossible (`or die(mysql_error())`).

```
mysql_query($query_rsRappelNiveaux, $connexionSport)
```

- Utilisez une fonction de transformation (voir le détail des différentes fonctions dans le tableau 18-1) de la ligne courante du jeu d'enregistrements afin de pouvoir exploiter les valeurs des résultats à l'aide d'un tableau de variables.

```
mysql_fetch_assoc($rsRappelNiveaux);
```

- Utilisez la syntaxe du tableau de variables adaptée à la fonction utilisée précédemment pour afficher la valeur d'un champ spécifique du jeu d'enregistrements (voir repère 2 de la figure 18-13).

```
echo $row_rsRappelNiveaux['niveau'];
```

- À la fin de la page dynamique, utilisez la fonction `mysql_free_result()` afin de libérer la mémoire utilisée par le jeu d'enregistrements.

```
mysql_free_result($rsRappelNiveaux);
```

- Enregistrez votre fichier. Pour tester le fonctionnement de la page dynamique, vous pouvez utiliser la fonction Live Data de Dreamweaver ou consulter votre page dans le Web local de Wamp 5. Pour utiliser le mode Live Data de Dreamweaver, passez en mode création (voir repère 1 de la figure 18-14), puis cliquez sur le bouton Live Data situé à droite des trois boutons de sélection de mode (voir repère 2 de la figure 18-14). La valeur du premier niveau enregistré dans la base cours (soit débutant) doit alors s'afficher dans l'éditeur (voir repère 3 de la figure 18-14).

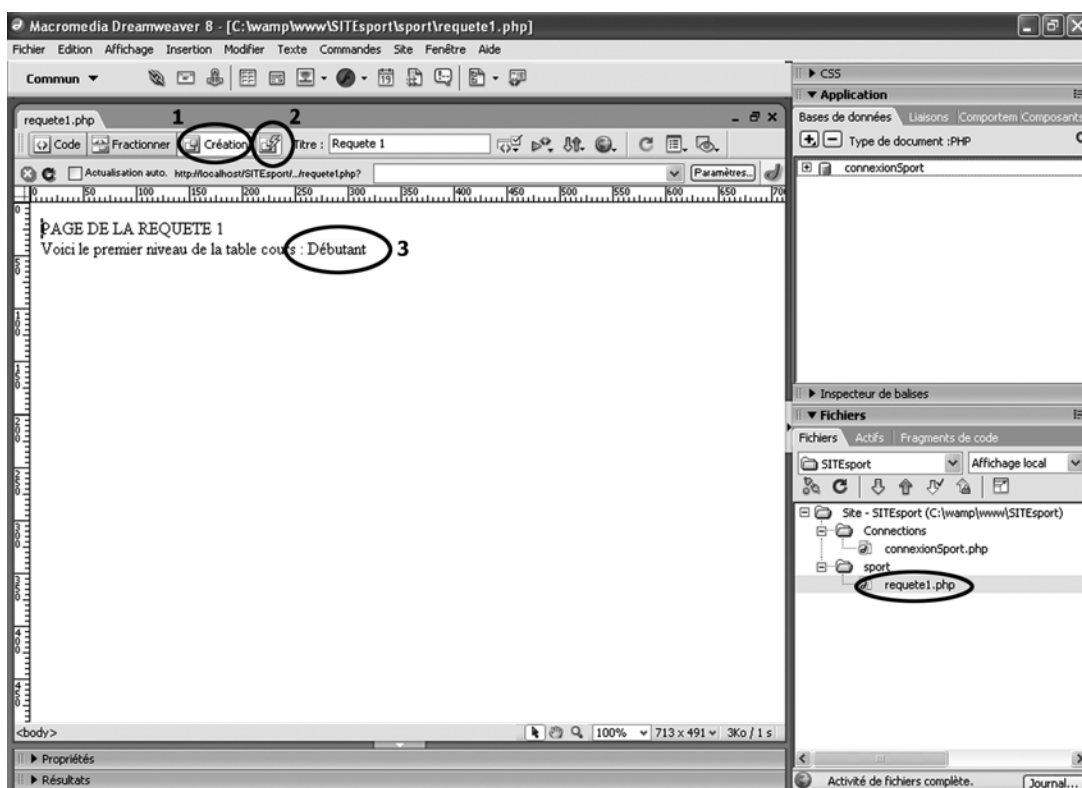


Figure 18-14

Test de la page dynamique `requete1.php` à l'aide de la fonction Live Data de Dreamweaver

#### À noter

Le mode Live Data ne peut fonctionner que si vous avez configuré votre serveur d'évaluation comme nous l'avons indiqué au début de ce chapitre.

10. Pour tester votre page dans le Web local de Wamp 5, ouvrez un navigateur en Web local (déroulez le menu du manager de Wamp 5 et sélectionnez localhost) puis parcourez le répertoire du site SITEsport jusqu'à la page `requete1.php`. La valeur du premier niveau de cours doit s'afficher dans la page de la même manière qu'en Live Data (voir figure 18-15).

Script de création et de mise en forme d'un jeu d'enregistrements (voir repère 1 de la figure 18-13) :

```
<?php
require_once('../Connections/connexionSport.php');
// Permet de disposer de la variable $database_connexionSport et
// de l'identificateur de connexion "$connexionSport" créé
// dans le fichier externe "connexionSport.php"
mysql_select_db($database_connexionSport, $connexionSport);
// Sélection de la base sport_db mémorisée
// dans la variable $database_connexionSport
$query_rsRappelNiveaux = "SELECT niveau FROM cours";
// Rédaction de la requête SQL
// dans la variable $query_rsRappelNiveaux
$rsRappelNiveaux = mysql_query($query_rsRappelNiveaux, $connexionSport) or die(mysql_error());
// Soumission de la requête et récupération
// du pointeur du jeu d'enregistrements
// dans la variable $rsRappelNiveaux
$row_rsRappelNiveaux = mysql_fetch_assoc($rsRappelNiveaux);
// Transformation de la première ligne
// du jeu d'enregistrements dans un tableau associatif
?>
```

Script d'utilisation d'un élément du jeu d'enregistrements (voir repère 2 de la figure 18-13) :

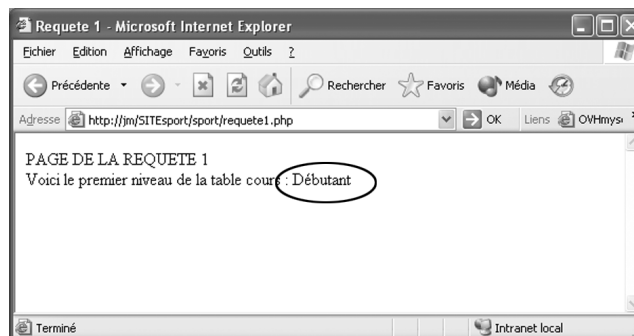
```
<?php
echo $row_rsRappelNiveaux['niveau'];
// Affiche la valeur du champ nom
// de la première ligne du jeu d'enregistrements
?>
```

Script de libération de la mémoire utilisée par le jeu d'enregistrements (voir repère 3 de la figure 18-13) :

```
<?php
mysql_free_result($rsRappelNiveaux);
// Libère la mémoire utilisée par la requête $rsRappelNiveaux
?>
```

**Figure 18-15**

*Test de la page  
dynamique `requete1.php`  
dans le Web local  
de Wamp 5*



## Exploitation d'un jeu d'enregistrements

Après la soumission de la requête à la base de données, celle-ci renvoie un pointeur qui ne peut pas être exploité directement. En effet, les résultats renvoyés par une base de données peuvent être conséquents et il n'est pas concevable d'enregistrer systématiquement la totalité de ces informations. Il faut donc utiliser une fonction de récupération des résultats afin de parcourir chacune des lignes du jeu d'enregistrements.

Ces différentes fonctions de récupération (de type `mysql_fetch_xxx`) sont présentées dans le tableau 18-1. Dans l'exemple précédent, nous en avons d'ailleurs utilisé une pour récupérer le champ nom de la première ligne du jeu d'enregistrements. Cependant, en pratique, vous aurez souvent besoin d'exploiter les résultats des différentes lignes du jeu d'enregistrement. Nous vous proposons donc d'étudier deux techniques de récupération des résultats issus d'un jeu d'enregistrements (par affichage direct à l'écran ou par transfert dans un tableau de variables).

### Affichage des différentes lignes d'un jeu d'enregistrements

Cette première technique permet d'afficher toutes les lignes d'un jeu d'enregistrements dans une page dynamique. Pour chaque ligne, seuls certains champs du jeu d'enregistrements seront affichés (voir figure 18-17). Pour illustrer cette technique, nous afficherons la liste des différents niveaux de cours en précisant le jour et l'horaire correspondants. Le script sera enregistré dans un fichier `requete2.php` dans le même répertoire `sport/` que le premier script.

Pour dérouler toutes les lignes du jeu d'enregistrements, utilisez une boucle `while` dont la condition de boucle sera réalisée par l'exécution d'une fonction `mysql_fetch_asso()`. Ainsi, dès que vous atteindrez la dernière ligne du jeu d'enregistrements, la fonction `mysql_fetch_asso()` retournera une valeur nulle, stoppant automatiquement la boucle `while` et du même coup l'affichage des lignes du jeu d'enregistrements.

#### À noter

Dans le cas d'un affichage partiel des champs de chaque ligne, il est plus judicieux d'utiliser une fonction `mysql_fetch_asso()` — qui permet de récupérer chaque valeur des champs à l'aide de son nom dans un tableau associatif (exemple : `$row_rs['nom']`) — qu'une fonction `mysql_fetch_row()` — pour laquelle il faut connaître l'indice correspondant à chaque champ (exemple : `$row_rs[2]`).

Script de création du jeu d'enregistrements (voir repère 1 de la figure 18-16) :

```
<?php
require_once('../Connections/connexionSport.php');
mysql_select_db($database_connexionSport, $connexionSport);
$query_rsListeCours = "SELECT * FROM cours";
$rsListeCours = mysql_query($query_rsListeCours, $connexionSport) or die(mysql_error());
?>
```

Script d'affichage du jeu d'enregistrements (voir repère 2 de la figure 18-16) :

```
<?php
// Affichage de la liste des cours sur une ligne différente
```

```

while($row_rsListeCours = mysql_fetch_assoc($rsListeCours)) {
echo "Le cours de niveau <b>".$row_rsListeCours['niveau']. "</b>";
echo " a lieu le <b>".$row_rsListeCours['jour']. "</b>";
echo " à <b>".$row_rsListeCours['horaire']. "</b> heure <br>";
}
?>

```

Script de libération de la mémoire utilisée par le jeu d'enregistrements (voir repère 3 de la figure 18-16) :

```

<?php
mysql_free_result($rsListeCours);
?>

```

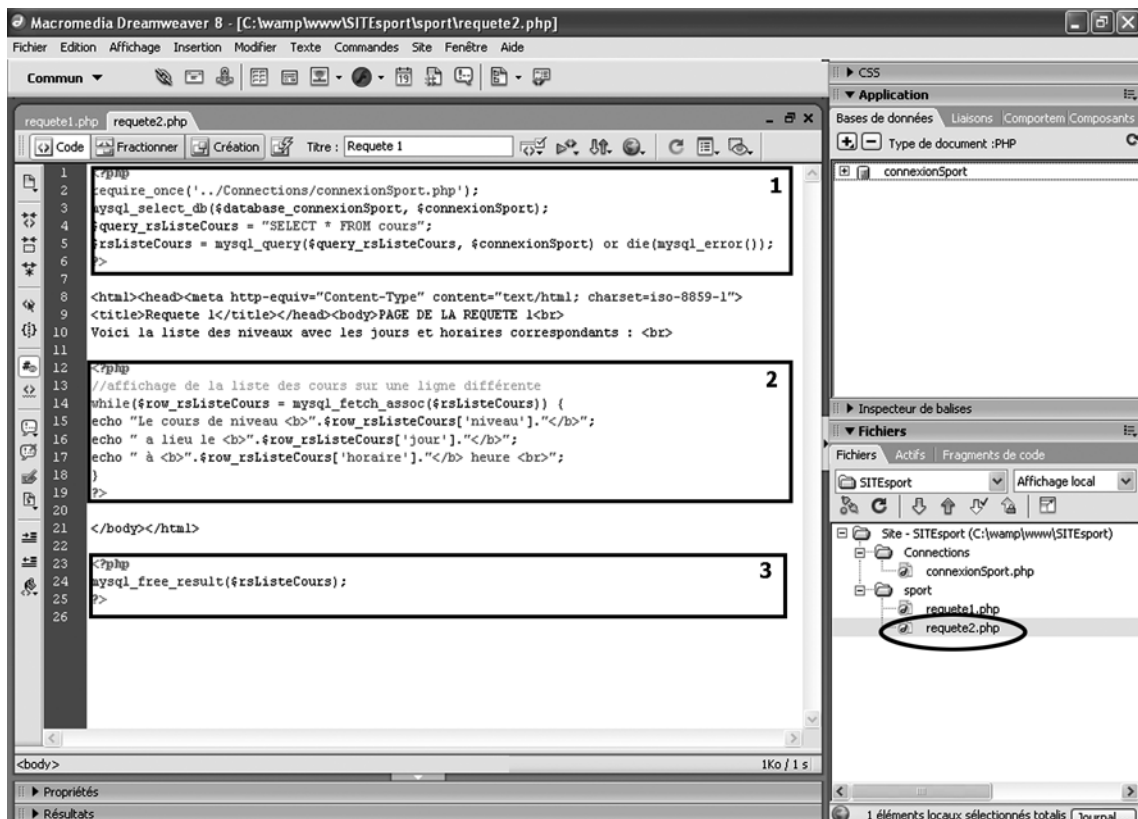


Figure 18-16

Script de la page dynamique *requete2.php*

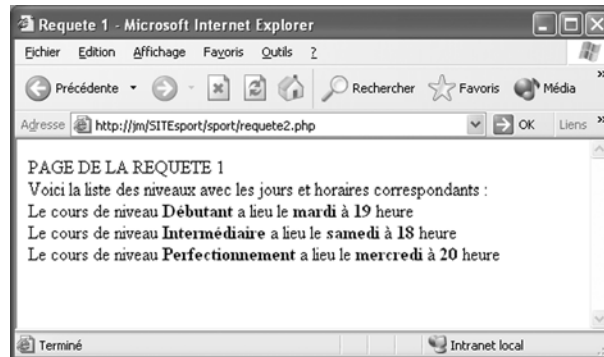


Figure 18-17

Test de la page dynamique *requete2.php* dans le Web local de Wamp 5

### Transfert d'un jeu d'enregistrements dans un tableau de variables

Cette seconde technique permet de récupérer toutes les valeurs retournées par la base de données en les enregistrant dans un tableau de variables à deux dimensions. Pour illustrer cette technique, nous l'appliquerons au transfert de la table *cours* dans un tableau `$tableauResultat[ ][ ]`.

Contrairement à l'exemple précédent, nous choisirons ici la fonction `mysql_fetch_array()`, qui permet de récupérer une ligne de résultat dans un tableau indicé (traitement à l'aide d'une boucle évoluant selon l'indice du tableau, par exemple).

Cette technique est très intéressante, car elle permet de transformer rapidement un jeu d'enregistrements en tableau de variables afin de traiter le résultat d'une requête.

Pour vérifier que les valeurs du tableau à deux dimensions correspondent bien à celles de la table *cours*, nous utiliserons la fonction `print_r()` pour afficher le résultat à l'écran (voir figure 18-19).

Script de création du jeu d'enregistrements (voir repère 1 de la figure 18-18) :

```
<?php
require_once('../Connections/connexionSport.php');
mysql_select_db($database_connexionSport, $connexionSport);
$query_rsListeCours = "SELECT * FROM cours";
$rsListeCours = mysql_query($query_rsListeCours, $connexionSport) or die(mysql_error());
?>
```

Script de transfert du jeu d'enregistrements dans le tableau à deux dimensions (voir repère 2 de la figure 18-18) :

```
<?php
$i=0;//init de l'indice du tableau
while($row_rsListeCours = mysql_fetch_row($rsListeCours)) {
    $tableauResultat[$i]=$row_rsListeCours;
```

```
//affectation d'une ligne d'enregistrement
$i++;//incréméntation de l'indice du tableau
}
?>
```

Script d'affichage du jeu d'enregistrements (voir repère 3 de la figure 18-18) :

```
<?php
echo "<pre>";
print_r($tableauResultat);
echo "</pre>";
echo '<br> Pour exemple, voici la valeur de $tableauResultat[2][3] = '.$tableauResultat[2][3];
?>
```

Script de libération de la mémoire utilisée par le jeu d'enregistrements (voir repère 4 de la figure 18-18) :

```
<?php
mysql_free_result($rsListeCours);
?>
```

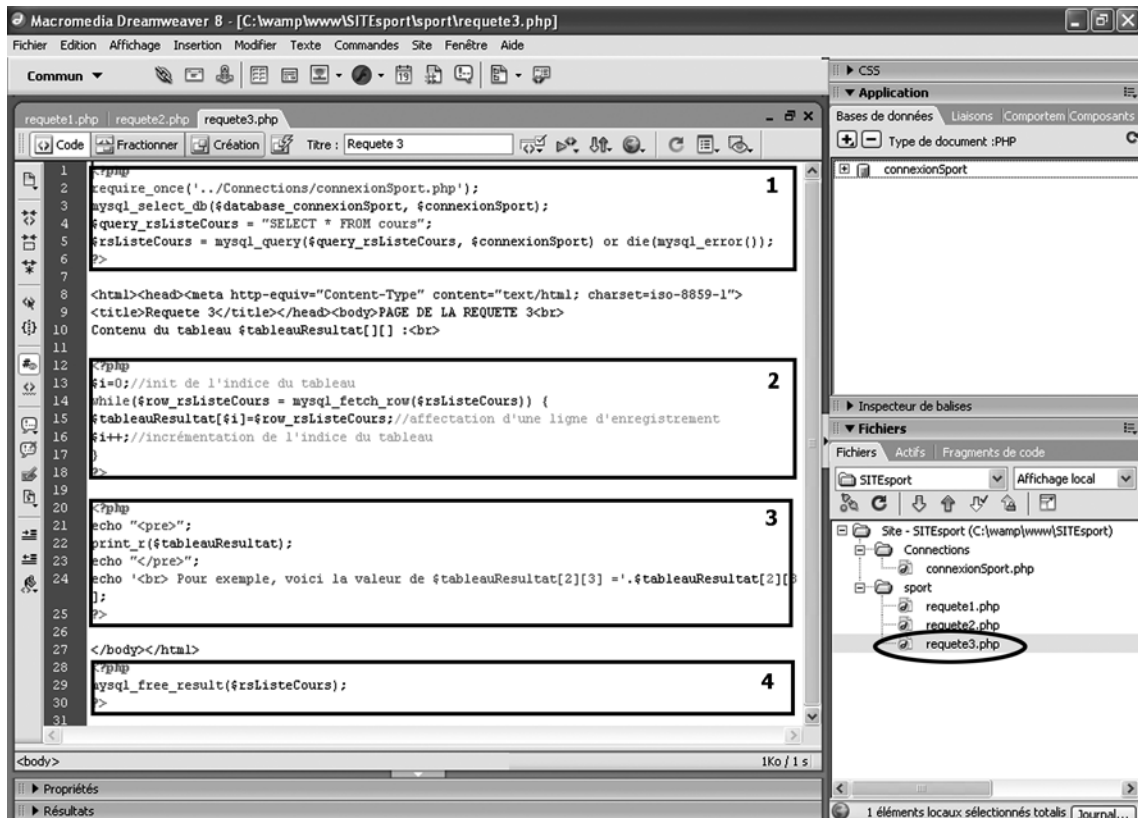


Figure 18-18

Script de la page dynamique `requete3.php`

```
Requete 3 - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Recherche Favoris
Adresse http://jm/SITEsport/sport/requete3.php OK Liens
PAGE DE LA REQUETE 3
Contenu du tableau $tableauResultat[][] :

Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => Débutant
            [3] => mardi
            [4] => 19
        )

    [1] => Array
        (
            [0] => 2
            [1] => 1
            [2] => Intermédiaire
            [3] => samedi
            [4] => 18
        )

    [2] => Array
        (
            [0] => 3
            [1] => 1
            [2] => Perfectionnement
            [3] => mercredi
            [4] => 28
        )
)

Pour exemple, voici la valeur de $tableauResultat[2][3]=mercredi
```

**Figure 18-19**

Test de la page dynamique *requete3.php* dans le Web local de Wamp 5

## Construction d'une requête SQL

Dans les exemples présentés ci-dessus, nous avons utilisé une requête SQL non paramétrable (exemple : `$query="SELECT * FROM adherents"`). Cependant, en pratique vous aurez souvent à créer une requête SQL dont les éléments sont issus de variables envoyées par un formulaire, dans l'URL ou encore par le biais d'une application Flash. Voici deux techniques de personnalisation d'une requête SQL (à partir d'une variable obligatoire ou de variables optionnelles).

### Construction d'une requête à partir d'une variable obligatoire

Cette première technique permet d'afficher la liste des adhérents dont l'année de naissance est supérieure à une année donnée (voir figure 18-20). Pour réaliser cette petite application, nous allons d'abord créer un formulaire HTML (*formulaire4.htm*) dans lequel sera intégré un champ de saisie

nommé `anneeNaissance`. Ce formulaire (voir figure 18-20) sera configuré pour envoyer ses données en méthode POST à un script `requete4.php` que nous réaliserons dans un second temps.

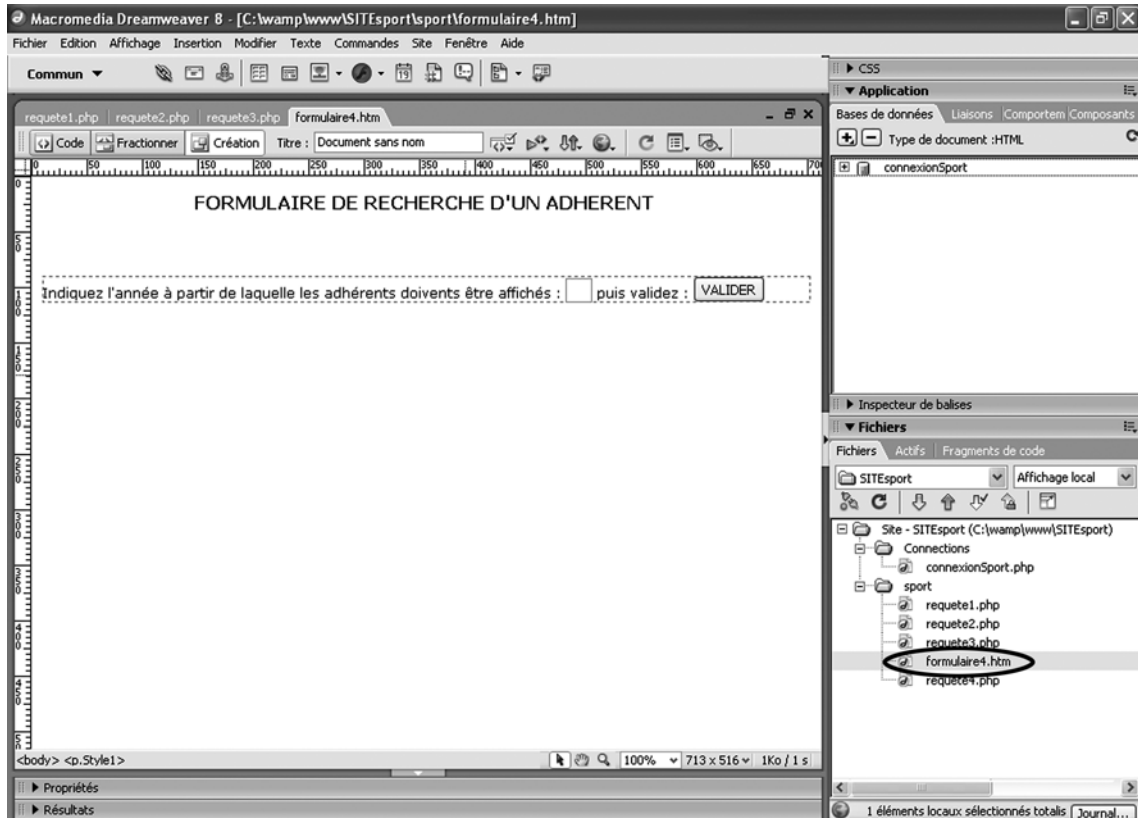


Figure 18-20

Formulaire de saisie de l'année de naissance : `formulaire4.htm`

Le script PHP `requete4.php` réceptionne la variable envoyée par le formulaire. Cette variable personnalise la clause `WHERE` de la requête SQL pour sélectionner uniquement les adhérents dont l'année de naissance est supérieure à la valeur transmise (soit `$anneeNaissance`). La première ligne du script permet d'initialiser la variable `$anneeNaissance` avec la valeur de la variable HTTP correspondante transmise par la méthode POST (`$_POST['anneeNaissance']`).

Les deux lignes qui suivent permettent d'établir une connexion avec le serveur MySQL et de sélectionner la base `sport_db` :

```
require_once('../Connections/connexionSport.php');
mysql_select_db($database_connexionSport, $connexionSport);
```

**À noter**

L'instruction d'initialisation est conditionnée par un test qui détermine si la variable HTTP existe (`isset()`). Dans le cas contraire, la variable `$anneeNaissance` est initialisée avec une valeur par défaut (1900 dans notre exemple) afin d'éviter l'affichage à l'écran d'un message signalant que la variable est indéfinie (voir repère 1 de la figure 18-21).

```
if(isset($_POST['anneeNaissance'])) $anneeNaissance=$_POST['anneeNaissance'];
else $anneeNaissance="1900";
```

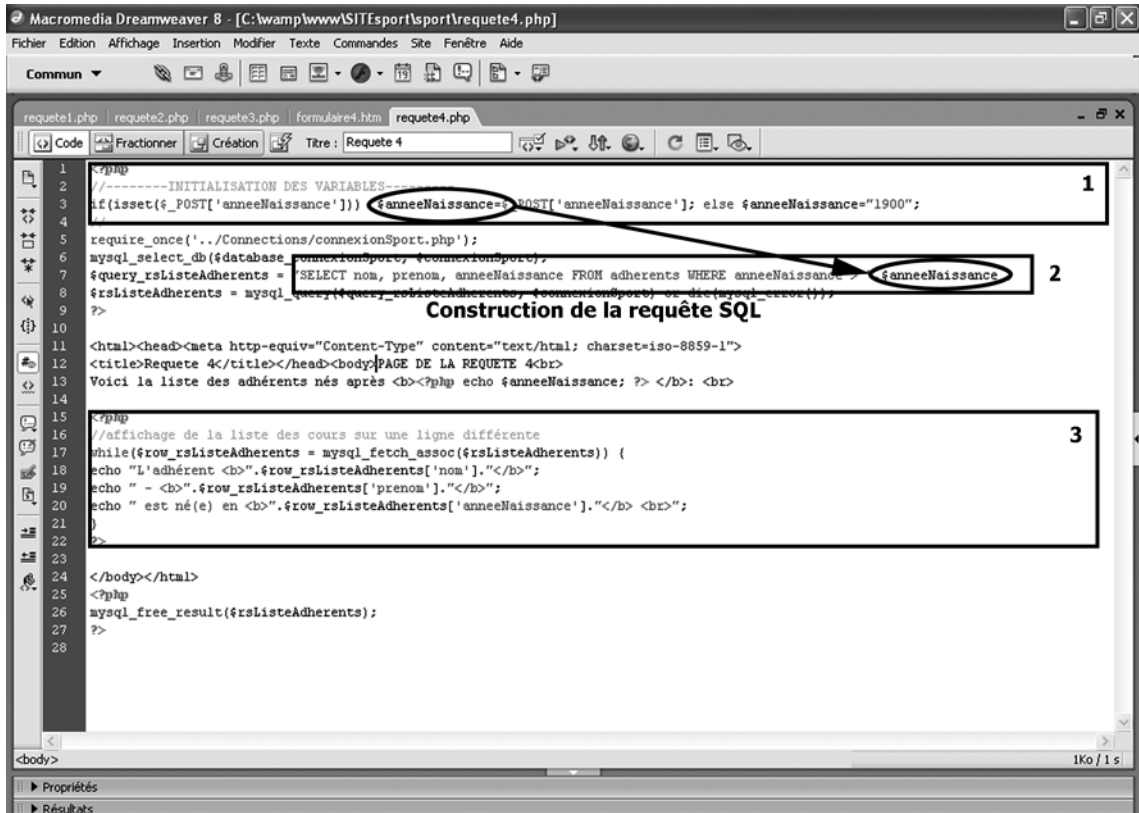


Figure 18-21

Script de la page dynamique `requete4.php`

La création de la requête est réalisée par concaténation de la variable `$anneeNaissance` avec le début de la requête (voir repère 2 de la figure 18-21). Lors de l'exécution du script, la requête est reconstituée avant d'être transmise au serveur MySQL via la fonction `mysql_query()` :

```
$query_rsListeAdherents = "SELECT nom, prenom, anneeNaissance FROM adherents WHERE
↳ anneeNaissance > ".$anneeNaissance ;
$rsListeAdherents = mysql_query($query_rsListeAdherents, $connexionSport) or
↳ die(mysql_error());
```