

## Festival : Découverte de la gestion des établissements

### Description du scénario

Propriétés	Description
<b>Intitulé long</b>	<i>Découverte de la gestion des établissements dans le contexte Festival,</i>
<b>Présentation</b>	Comprendre et vérifier la réalisation d'un cas d'utilisation. Concevoir une amélioration du comportement et une meilleure communication.
<b>Notions</b>	<i>C.1.2 Formalisation des besoins Expression des besoins : cas d'utilisation, jeu d'essai C.2.3. Programmation des traitements Procédure, fonction, paramètre, structure de contrôle, mise au point de l'application.</i>
<b>Transversalité</b>	
<b>Pré-requis</b>	notion de variable, structure conditionnelle, structure itérative, le langage PHP,
<b>Objectifs</b>	Évaluer la réalisation d'un cas d'utilisation, comprendre le fonctionnement de l'application Festival,
<b>Outils</b>	Côté client : navigateur Côté serveur : simple éditeur, apache, php, mysql,
<b>Mots-clés</b>	Cas d'utilisation, PHP, accès à la base de donnée, contraintes d'intégrité SQL.
<b>Durée</b>	2 heures + 2 heures pour la partie évolution,
<b>Auteur(es)</b>	Olivier Capuozzo, Patrice Grand
<b>Version</b>	v 1.0
<b>Date de publication</b>	février 2007

### Énoncé

« Le festival de musique *Folklores du Monde* est organisé la première semaine de juillet de chaque année dans la ville de *Saint-Malo*. Il permet de réunir des groupes étrangers ainsi que quelques groupes français.» :

<http://www.reseaucerta.org/cotecours/pub.php?num=390>

L'écran suivant permet de gérer les établissements d'hébergement des groupes :

Etablissements			
Collège Lamartine	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	
Collège de Moka	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	
Collège Léonard de Vinci	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	
Centre de rencontres internationales	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
<a href="#">Création d'un établissement</a>			

La suppression d'un établissement est décrite dans l'extrait de cas d'utilisation suivant :

### 2.c. Supprimer un établissement

- 2.c.1. L'agent sélectionne la commande de suppression d'un établissement
- 2.c.2. Le système retourne une demande de confirmation de suppression.
- 2.c.3. L'agent confirme.
- 2.c.4. Le système supprime définitivement l'établissement de ses données.
- 2.c.5. Retour en 1.

#### Cas particuliers

- 2.c.3. L'agent ne confirme pas :
  - retour en 1.
- 2.c.4. Des attributions sont actuellement concernées par l'établissement en question :
  - 2.c.4.1. Le système ne supprime pas l'établissement.
  - 2.c.4.2. Retour en 1.

## Travail à Réaliser

Lancer l'application Festival, sélectionner la gestion des établissements.

1. Expliquez comment se traduit dans l'application la demande de confirmation de suppression (point 2.c.2)
2. Expliquez comment se traduit dans l'application le retour en 1 (point 2.c.5)
3. Pourquoi les établissements faisant l'objet d'attributions sont-ils traités dans un cas particulier (point 2.c.4) ?
4. Quelle contrainte sur la base de données ne serait pas respectée alors ?
5. Comment se traduit dans l'application le cas particulier 2.c.4 ?

Le fichier mettant en œuvre ce traitement est le fichier : *listeEtablissements.php*

Sa logique peut être résumée aux étapes suivantes :

```
$connexion=connect();
selectBase($connexion);
$req=obtenirReqEtablissements();

$rsEtab=mysql_query($req, $connexion);

$lgEtab=mysql_fetch_array($rsEtab);
while ($lgEtab!=FALSE) {
    $id=$lgEtab['id'];
    $nom=$lgEtab['nom'];

    if ( ! existeAttributionsEtab($connexion, $id) ) {
        // mis en place du lien de suppression
    }

    $lgEtab=mysql_fetch_array($rsEtab);
}
```

Diagram illustrating the logic of the `listeEtablissements.php` file, with callouts explaining the steps:

- connexion et sélection de la base de données
- obtenir le texte de la requête SQL
- exécution de la requête SQL
- prendre la première ligne
- pour chaque ligne du résultat faire ...
- passer à la ligne suivante

6. Quel est le nom de la fonction utilisée pour déterminer la présence ou non du lien de suppression ?

La fonction se situe dans le module *\_gestionBase.inc.php* (ligne 172).

```
// Teste la présence d'attributions pour l'établissement transmis
function existeAttributionsEtab($connexion, $id)
{
    $req="select * From Attribution where idEtab='$id'";
    $rsAttrib=mysql_query($req, $connexion);
    return mysql_fetch_array($rsAttrib);
}
```

7. Quel traitement réalise la fonction ?
8. En vous aidant de l'aide du langage sur le site <http://www.manuelphp.com/php/function.mysql-fetch-array.php>, indiquer le type de retour de la fonction `existeAttributionsEtab`.
9. Sachant qu'il existe en PHP une fonction nommée `mysql_num_rows`, qui retourne le nombre de lignes résultant d'une requête SQL de sélection, proposer une autre écriture de la fonction `existeAttributionsEtab`.  
Plus d'informations ici : <http://www.manuelphp.com/php/function.mysql-num-rows.php>
10. Écrire l'url pointée par le lien « supprimer » lorsque l'utilisateur sélectionne « Gestion des établissements ». Identifier les différentes parties de cette url.
11. Dans le navigateur (barre d'adresse) taper l'url concernant une demande de suppression pour un établissement ayant des attributions. Que constatez-vous ? Qu'en pensez-vous ?
12. Quels sont ses effets réels sur les données ? Justifier les actions ou non action constatées.

Évolution envisagée,

A la place de l'absence de lien de suppression, on souhaiterait voir apparaître le nombre d'attributions en cours, par exemple entre parenthèses.

Festival Folklores du Monde			
Hébergement des groupes			
<a href="#">Accueil</a>	<a href="#">Gestion établissements</a>	<a href="#">Attributions chambres</a>	
<b>Etablissements</b>			
Collège Lamartine	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	(21 attributions)
Collège de Moka	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	(20 attributions)
Collège Léonard de Vinci	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	(20 attributions)
Centre de rencontres internationales	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
<a href="#">Création d'un établissement</a>			

13 Écrire le code nécessaire pour réaliser cette modification, sachant que le nombre d'attributions peut être obtenu en appelant la fonction `obtenirNbOccup`. (voir la bibliothèque `_gestionBase.inc.php`.)

## Corrigé

1. Expliquez comment se traduit dans l'application la demande de confirmation de suppression (2.c.2)

L'application présente deux liens : un qui déclenche la suppression de l'établissement et un autre qui retourne à la liste des établissements.

En plaçant le curseur de la souris au dessus d'un lien, l'utilisateur peut observer la nature de la requête à destination du serveur+application dans la barre d'état, au bas de la fenêtre du navigateur.

**Souhaitez-vous vraiment supprimer l'établissement Centre de rencontres Internationales ?**

[Oui](#)   [Non](#)

2. Expliquez comment se traduit dans l'application le retour en 1 (2.c.5)

A la fin du traitement, l'utilisateur est redirigé vers l'onglet de gestion des établissements.

3. Pourquoi les établissements faisant l'objet d'attributions sont-ils traités dans un cas particulier (point 2.c.4) ?

La séquence principale d'un cas d'utilisation spécifie toujours un cas de succès, les alternatives sont reléguées dans la section « cas particulier »

4. Quelle contrainte sur la base de données ne serait pas respectée alors ?

Certaines valeurs de clés étrangères de la table Attributions ne seraient plus cohérentes si la ligne de la table Établissement qu'elles référencent était supprimée (contrainte d'intégrité référentielle)

5. Comment se traduit dans l'application le cas particulier 2.c.4 ?

La commande « supprimer » n'est disponible que pour les établissements ne contrariant pas la règle de gestion exprimée en 2.c.4 :

Etablissements			
Collège Lamartine	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	
Collège de Moka	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	
Collège Léonard de Vinci	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	
Lycée Jean Jaurès	<a href="#">Voir détail</a>	<a href="#">Modifier</a>	<a href="#">Supprimer</a>

6. Quel est le nom de la fonction utilisée pour déterminer la présence ou non du lien de suppression ?

C'est la fonction `existeAttributionsEtab`, utilisée avec l'opérateur `!` (not).

```
if ( ! existeAttributionsEtab($connexion, $id) )  
et équivalent à  
if ( FALSE == existeAttributionsEtab($connexion, $id) )
```

7. Quel traitement réalise la fonction existeAttributionsEtab ?

```
// Teste la présence d'attributions pour l'établissement transmis
function existeAttributionsEtab($connexion, $id)
{
    $req="select * From Attribution where idEtab='$id'";
    $rsAttrib=mysql_query($req, $connexion);
    return mysql_fetch_array($rsAttrib);
}
```

La fonction effectue une requête SQL de sélection sur la base de données, afin de vérifier l'existence d'une occurrence d'une ligne répondant à un critère basé sur une valeur de clé primaire.

Cette fonction prend en argument deux variables : \$connexion et \$id  
\$connexion : identifiant de la connexion à la base de données, valorisé en début de script.  
\$id : identifiant de l'établissement en cours (c'est à dire de la ligne HTML en construction – nous sommes dans une boucle)

8. En vous aidant de l'aide du langage sur le site <http://www.manuelphp.com/php/function.mysql-fetch-array.php>, indiquer le type de retour de la fonction existeAttributionsEtab.

La valeur de retour est déterminée par l'expression à droite de return :

```
return mysql_fetch_array($rsAttrib);
```

Pour comprendre le type de retour de cette fonction, il faut comprendre les spécifications de la fonction et l'expression de la valeur de vérité de PHP.

1/ Extrait des spécifications de la fonction :

<http://www.manuelphp.com/php/function.mysql-fetch-array.php>

*array **mysql\_fetch\_array** (resource result)*

***mysql\_fetch\_array** retourne un tableau qui contient la ligne demandée dans le résultat result et déplace le pointeur de données interne d'un cran, ou retourne FALSE s'il n'y a plus de lignes.*

On comprend que le résultat **ne sera pas** FALSE si l'appel à la fonction *mysql\_fetch\_array* retourne un tableau.

2/ PHP considère comme TRUE toute valeur n'étant pas associée à FALSE. Ainsi un tableau contenant des valeurs sera associé à la valeur TRUE.

Plus précisément : <http://www.manuelphp.com/php/language.types.boolean.casting.php>  
les valeurs suivantes sont considérées comme fausses ( FALSE ) :

- Le booléen FALSE lui-même
- L' entier 0 (zéro)
- La chaîne de caractères 0.0 (zéro)
- La chaîne de caractères vide et la chaîne de caractères "0"
- Le tableau vide (aucun élément)
- L' objet vide (aucun élément) (PHP 4 uniquement)
- Le type spécial NULL

9. Sachant qu'il existe en PHP une fonction nommée mysql\_num\_rows, qui retourne le nombre de lignes résultant d'une requête SQL de sélection, proposer une autre écriture de la fonction existeAttributionsEtab.

Plus d'informations ici : <http://www.manuelphp.com/php/function.mysql-num-rows.php>

L'utilisation d'une fonction qui nous donnerait le nombre de lignes retournées.

*int mysql\_num\_rows ( resource result )*

*Récupère le nombre de lignes d'un jeu de résultat. Cette commande n'est disponible que pour les requêtes comme SELECT*

```
// Teste la présence d'attributions pour l'établissement transmis
function existeAttributionsEtab($connexion, $id)
{
    $req="select * From Attribution where idEtab='$id'";
    $rsAttrib=mysql_query($req, $connexion);
    return mysql_num_rows($rsAttrib) > 0;
}
```

*Autre solution, utiliser une requête qui compte le nombre de lignes vérifiant la condition.*

```
// Teste la présence d'attributions pour l'établissement transmis
function existeAttributionsEtab($connexion, $id)
{
    $req = "select count(*) as nbLignes From Attribution where
            idEtab='$idEtab'";
    $rsAttrib = mysql_query($req, $connexion);
    $rs = mysql_fetch_array($rsAttrib);
    return $rs['nbLignes'] > 0;
}
```

#### REMARQUE :

Il est peu probable que la majorité des élèves trouvent une solution aussi concise que :

*return mysql\_num\_rows(\$rsAttrib) > 0;*

Nous avons imaginé ci-dessous quelques solutions commentées, pouvant être produites par les élèves.

Solutions produisant un résultat correct	Commentaires
<pre>(A) if ( mysql_num_rows(\$rsAttrib) &gt; 0) == TRUE) {     return TRUE; } if ( mysql_num_rows(\$rsAttrib) == 0) == TRUE) {     return FALSE; }</pre>	voir ci-dessous
<pre>(B) if ( mysql_num_rows(\$rsAttrib) &gt; 0) == TRUE) {     return TRUE; } if ( mysql_num_rows(\$rsAttrib) &gt; 0) == FALSE) {     return FALSE; }</pre>	Ces deux propositions ne tiennent pas compte que les conditions sont liées : si la première condition est vérifiée alors l'autre ne peut l'être et inversement. les - : <ul style="list-style-type: none"><li>• Lecture difficile du code car celui-ci prête à croire que les conditions ne sont pas liées,</li><li>• Traitement inutile ; dans le cas où la première condition est vérifiée, le système évaluera toujours la deuxième condition.</li></ul>
<pre>(C) if (mysql_num_rows(\$rsAttrib) &gt; 0 == TRUE) {     return TRUE; } else {     return FALSE; }</pre>	voir ci-dessous

```

(D)
if (mysql_num_rows($rsAttrib) > 0) {
    return TRUE;
}
else {
    return FALSE;
}

```

les versions (C) et (D) sont parfaitement équivalentes.  
 Dans (C); il est inutile de préciser == TRUE car c'est le comportement par défaut de la structure de décision :  
 si (la condition est vérifiée) alors ... sinon ....

Le + : l'élève a compris que le résultat à la question 'est-ce qu'un nombre est supérieur à zéro ?' est soit VRAI soit FAUX. (utilisation du **else**)  
 Le - : le code semble traduire la valeur de l'expression entre parenthèses par une autre valeur, or il n'en n'est rien. (c'est une erreur assez répandue).

... plus toutes les variantes possibles et imaginables, à l'infini, consistant à utiliser des variables locales pour stocker des valeurs intermédiaires (par exemple pour éviter plusieurs appels inutiles à la même fonction, ou la volonté de n'avoir qu'une seule instruction *return*).

10. Écrire l'url pointée par le lien « supprimer » proposé à l'utilisateur lorsqu'il sélectionne « Gestion des établissements ». Identifier les différentes parties de cette url.

<http://192.168.0.4/codeAppliFestival/suppressionEtablissement.php?action=demanderSupprEtab&id=11111111>

- http:// le protocole
- 192.168.0.4 : nom du domaine ou adresse IP
- codeAppliFestival/suppressionEtablissemen.php : localisation de la page Web demandée
- ? : caractère spécifiant que des données suivent
- action=demanderSupprEtab&id=0350785 : chaîne de requête, paramètres passés dans l'url, soit un identifiant d'action à lancer ainsi que l'identifiant d'un établissement.

plus d'information : <http://fr.wikipedia.org/wiki/URL>

11. Dans le navigateur (barre d'adresse) taper l'url concernant une demande de suppression pour un établissement ayant des attributions. Que constatez-vous ? Qu'en pensez-vous ?

En recopiant l'url de suppression, changeant l'identifiant de l'établissement et en soumettant la requête au serveur. Exemple :

<http://localhost/~kpu/codeAppliFestival/suppressionEtablissement.php?action=demanderSupprEtab&id=0350785N>

L'application a l'air de se comporter normalement, et s'apprête à supprimer l'établissement proposé (ici Collège de Moka), la confirmation est demandée à l'utilisateur.



suppressionEtablissement.php n'effectue aucun test de validité des valeurs des paramètres reçus.

12. Quels sont ses effets réels sur les données ? Justifier les actions ou non action constatées.

Aucune donnée n'est supprimée. Le système de gestion de la base de données interdit par défaut les suppressions d'une ligne référencée par d'autres (contrainte d'intégrité gérée par le SGBD).

Prolongement possible : effectuer la vérification afin de rendre le comportement logique face à ce type d'actions (suppression non autorisée d'établissements, consultation/modification d'un établissement inexistant, valeur d'id non renseignée). L'élève pourra utiliser une fonction existante de `_gestionBase.inc.php` ( ***estUnIdEtablissement*** ). Exemple :

```
if (estUnIdEtablissement($connexion, $id)) {  
    ...  
}  
else  
    ajouterErreur(" => Etablissement inexistant : " . $id);  
}
```

13. Présentation du nombre d'attributions. Écrire le code nécessaire pour réaliser cette modification, sachant que le nombre d'attributions peut être obtenu en appelant la fonction **obtenirNbOccup**. (voir la bibliothèque `_gestionBase.inc.php`.)

Étude de la page `listeEtablissements.php`

Les lignes 56 à 69 traitent du problème. En effet en ligne 58, une condition, basée sur l'appel de la fonction `existeAttributionsEtab`, permet de déterminer s'il faut inscrire un lien de suppression (ligne 62) ou rien du tout, représenté (ligne 68) par la présence du caractère d'espace insécable (`$nbsp`; - non-breaking space).

**Version actuelle**

```
// S'il existe déjà des attributions pour l'établissement, il faudra  
// d'abord les supprimer avant de pouvoir supprimer l'établissement  
if (!existeAttributionsEtab($connexion, $id)) {  
    echo "  
        <td width='16%' align='center'>  
        <a href='suppressionEtablissement.php?  
            action=demanderSupprEtab&id=$id'>  
            Supprimer</a></td>";  
} else {  
    echo "  
        <td width='16%'>&nbsp; </td>";  
}
```

Nous intervenons alors sur la ligne 68, afin de remplacer \$nbsp; par le nombre d'attributions alloués à l'établissement en cours, identifié par la variable `$id`. Pour cela il faut savoir si il n'existe pas dans la bibliothèque `_gestionBase.inc.php` une fonction dont c'est la tâche. On constate que la fonction nommée **obtenirNbOccup** paraît répondre parfaitement à notre attente. Observons sa signature (interface)

```
// Retourne le nombre de chambres occupées pour l'id étab transmis  
function obtenirNbOccup($connexion, $idEtab)
```

Nous pouvons donc l'utiliser car nous avons à notre disposition

- l'identification de connexion : la variable `$connexion` (déjà utilisée dans la page)
- l'identifiant de l'établissement en cours : `$id` (valorisé dans la boucle)

Nous placerons le résultat de l'appel à cette fonction entre parenthèses.

```
echo "<td width='16%'>&nbsp; </td>";
```

Devient

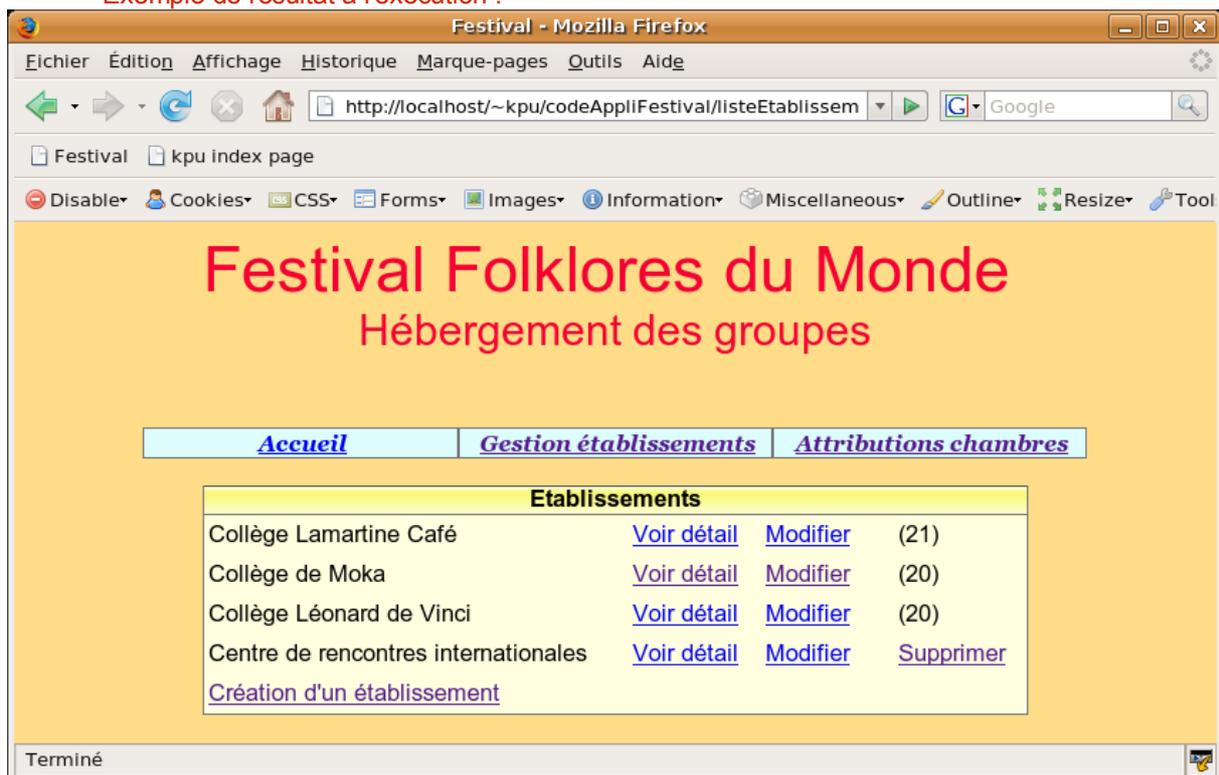
```
echo "<td width='16%'> (" . obtenirNbOccup($connexion, $id) . ")</td>";
```

On remarquera que la chaîne de caractères initiale "<td width='16%'>&nbsp; </td>"; a été découpée en trois parties afin d'inclure une valeur issue d'un appel de fonction. Les trois parties sont liées entre elles par un point, l'opérateur de concaténation de chaînes de caractères de PHP.

### ***Nouvelle Version (en gras le changement)***

```
// S'il existe déjà des attributions pour l'établissement, il faudra
// d'abord les supprimer avant de pouvoir supprimer l'établissement
if (!existeAttributionsEtab($connexion, $id)) {
    echo "
        <td width='16%' align='center'>
            <a href='suppressionEtablissement.php?
                action=demanderSupprEtab&amp;id=$id'>
                Supprimer</a></td>";
} else {
    echo "
        <td width='16%'> (" . obtenirNbOccup($connexion, $id) . ")</td>";
}
```

Exemple de résultat à l'exécution :



[Idée de prolongement possible de cette question :](#)

Le traitement de la présence ou non de la fonction de suppression pour un établissement donné mobilise 2 appels de fonction, **existeAttributionsEtab** et **obtenirNbOccup**, et donc au final deux requêtes SQL, alors qu'une seule requête portant sur le nombre d'attribution suffirait à résoudre le problème.

1. obtenir le nombre d'attributions actuelles de l'établissement (**obtenirNbOccup**) et stocker cette valeur dans une variable nommée \$nbAttrib.

2. si la valeur de la variable \$nbAttrib est égale à zéro alors on peut présenter le lien de suppression
3. sinon on affiche sa valeur.

Réalisation :

```
$nbAttrib = obtenirNbOccup($connexion, $id);
// S'il existe déjà des attributions pour l'établissement, il faudra
// d'abord les supprimer avant de pouvoir supprimer l'établissement
if ($nbAttrib == 0) {
    echo "
        <td width='16%' align='center'>
            <a href='suppressionEtablissement.php?
                action=demanderSupprEtab&amp;id=$id'>
                Supprimer</a></td>";
} else {
    echo "
        <td width='16%'> (" . $nbAttrib . ")</td>";
}
```

Suite possible : proposer d'inscrire (x attributions) ou (x attribution) avec gestion du singulier lorsque le nombre d'attribution est 1 (et non zéro car ce cas est traité par la présence du lien de suppression).

Réalisation

```
if ($nbAttrib ==1) {
    echo "<td width='16%'> (" . $nbAttrib . " attribution)</td>";
} else {
    echo "<td width='16%'> (" . $nbAttrib . " attributions)</td>";
}
```

Variante, sans répétition :

```
echo
"<td width='16%'>(" . $nbAttrib . " attribution".
(($nbAttrib == 1) ? "" : "s") . ")</td>";
```