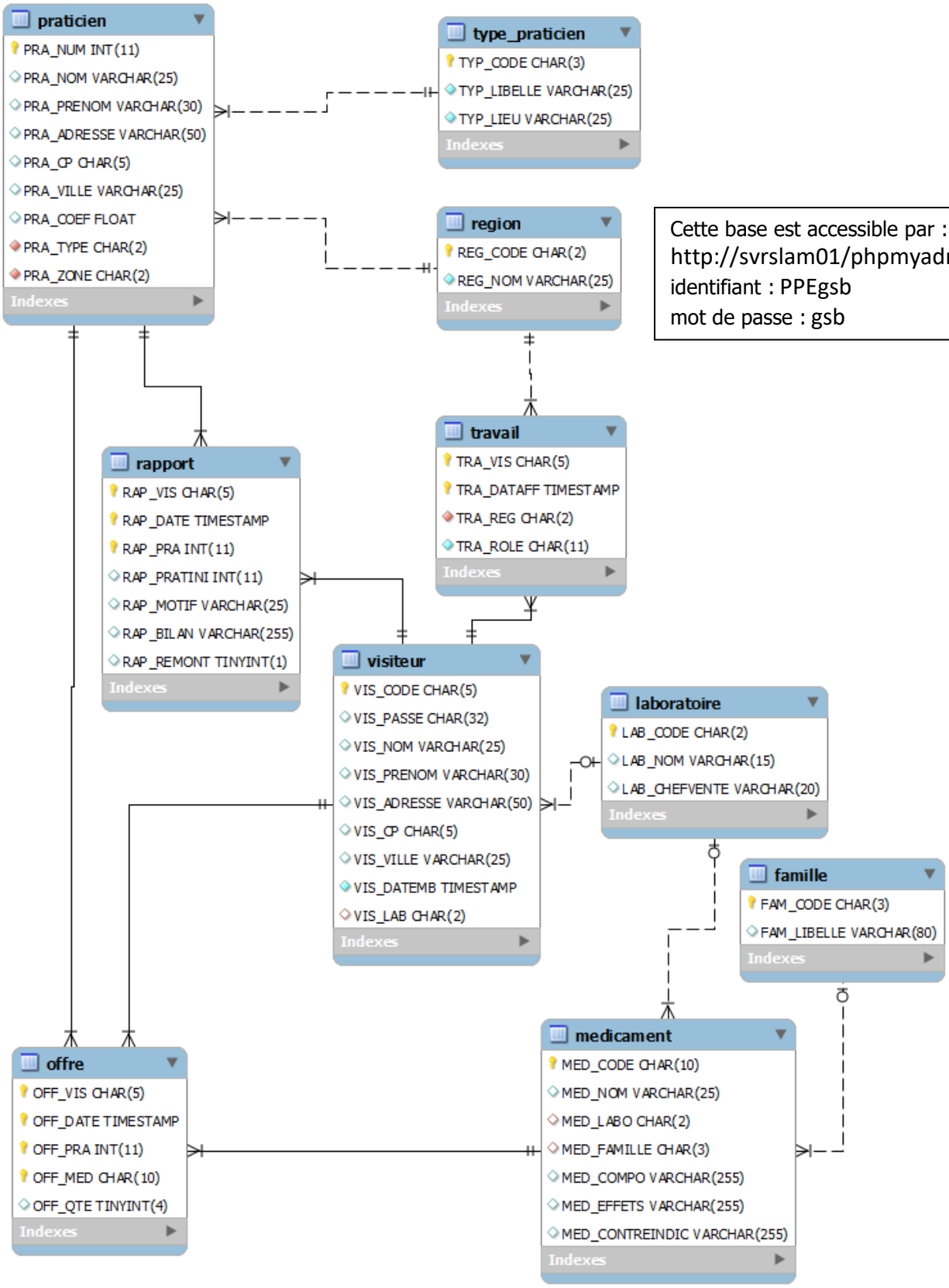
La base de données gsb



Le formulaire de consultation des praticiens

1 La base gsb

Elle a été créée sur le serveur MySQL situé sur svrslam01. Vous avez son schéma en page précédente. Elle est codée en UTF-8. Les pages web du site doivent également être encodées en UTF-8. Mais le serveur HTTP doit indiquer au SGBD MySQL que les échanges se font en UTF-8 à l'aide de la requête SQL suivante : `SET NAMES UTF8`
Cette requête doit être envoyée au SGBD juste après la connexion.

2 La connexion à la base de données.

Une classe de connexion est fournie avec le projet (`classConnexion.php`) : elle utilise ODBC. Nous allons plutôt travailler avec la classe PDO. Supprimez le fichier `classConnexion.php`.

Créer le fichier `SourceDonnees.inc.php` que vous pourrez inclure ensuite dans vos programmes. Dans ce fichier créez une fonction nommée `SGBDConnect()` qui assurera la connexion au SGBD en instanciant **et en retournant** un objet de la classe PDO (utilisateur **PPEgsb**, mot de passe **gsb** ; cet utilisateur possède le droit SELECT sur toutes les tables).

```
try {
   $connexion = new PDO('mysql:host=localhost;dbname=gsb', 'PPEgsb', 'gsb');
   $connexion->query('SET NAMES UTF8');
   $connexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo 'Erreur !: ' . $e->getMessage() . '<br />';
    exit();
}
```

Pour tester, ajoutez dans `index.php` un appel à votre fonction. Lancez ensuite le débogage du projet et exécutez pas à pas le code pour voir si la connexion s'effectue correctement. Supprimez le code ajouté.

3 La page formPRATICIEN

Ouvrez la page `formPRATICIEN.html` fournie. Cette page contient un script en langage **JavaScript**. C'est le seul langage de programmation implémenté dans tous les navigateurs. Il permet d'effectuer des traitements côté client.

Une page web peut comporter jusqu'à 3 aspects qu'il faut séparer :
- la structure (HTML),
- la présentation (CSS),
- les traitements (JavaScript).

Isolez le code JavaScript : ajoutez un dossier nommé `JavaScript` à votre projet. Créez dans ce dossier un fichier nommé `GCRAjax.js` et copiez-y le code JavaScript du fichier html (ensuite Source/Format vous épargnera du temps et des efforts pour reformater votre code).

Supprimez du fichier html le code et la balise qui l'encadre.

Le code étant isolé, vous pouvez maintenant :
- remplacer l'extension de cette page par `php` (en utilisant l'explorateur de Windows).
- Effacer tout le début de la page et faire appel à l'entête déjà créé.
- Supprimer tous les attributs de style.
- Dans les `<div>` remplacer l'attribut `name` par `id`.
- Faire appel au pied de page déjà créé.
- Mettre à jour le lien dans `entete.inc.php`.

Testez votre page. Dans le navigateur clic-droit/Code source de la page permet de vérifier le code HTML généré par votre programme. Votre page comporte deux formulaires dont un est vide.

3.1 Remplir la liste lstPrat avec tous les praticiens de la table.

Dans la liste déroulante `lstPrat` commencez par effacer le code `onClick="chercher(this.value);"` de la liste déroulante, puis ajoutez un bouton "Ok" qui permettra de soumettre ce formulaire.

Veiller dans tous les formulaires à utiliser l'attribut `tabindex` pour que le focus suive l'ordre des contrôles de saisie du formulaire.

Chaque ligne de la liste déroulante doit :
- afficher le nom et le prénom d'un praticien,
- avoir pour valeur (`value`) le numéro du praticien.

De cette façon, lorsque le formulaire sera soumis, c'est le numéro du praticien sélectionné qui sera récupéré pour être traité.

Remarque
La création d'une liste déroulante à partir d'un jeu d'enregistrements est une fonctionnalité très classique dans une application web. Aussi, plutôt que de créer son code puis de le dupliquer systématiquement, il est préférable de créer une fonction qui réalisera ce traitement à partir des paramètres qui lui seront fournis. Cette fonction réutilisable, que nous nommerons `formSelectDepuisRecordset()`, sera placée dans un fichier à inclure dans tous les programmes qui en auront besoin. D'autres fonctions réutilisables seront ajoutées dans le fichier au fur et à mesure des développements. Un tel fichier s'appelle une **bibliothèque** (de fonctions).

Pour développer ce genre de fonctionnalité, commencez par la manipulation des données, et terminez par l'interface utilisateur. Donc ici :

3.1.1 Etape 1

Dans PHPMyAdmin ou dans MySQL Workbench mettez au point la requête SQL qui retourne la liste des praticiens dont vous avez besoin.
Développez dans `SourceDonnees.inc.php` une fonction (nommée `getListePraticiens()`) qui récupère et retourne la liste des praticiens. Les praticiens doivent être triés par ordre alphabétique. N'oubliez pas le `try catch` pour l'accès à la base de données.

3.1.2 Etape 2

3.1.2.1 Cahier des charges de la fonction formSelectDepuisRecordset()

La fonction doit retourner le code HTML de création d'une liste déroulante avec son label. Les différents attributs de cette liste déroulante devront être paramétrables. La fonction doit donc recevoir en paramètre :
- le label de la liste déroulante,
- le nom de la liste déroulante,
- l'id de la liste déroulante,
- un jeu d'enregistrements (*recordset*) contenant les couples : valeur de l'option/libellé de l'option.
- la valeur de l'option sélectionnée (à faire ajouter dans une 2ème phase ?)
- l'index de tabulation de la liste déroulante.

Créez un fichier nommé `Bibliotheque01.inc.php` (à vous de voir où). Développez la fonction `formSelectDepuisRecordset()`.

Un problème : ici le libellé de l'option sera constitué du nom et du prénom du praticien. Avec le numéro de praticien, cela nous fait un jeu d'enregistrements comportant 3 colonnes, alors que la fonction attend un jeu d'enregistrements comportant 2 colonnes (code/libellé) ! Comment résoudre ce problème ?
La fonction `concat()` de MySQL devrait vous aidez ... Consultez sa documentation.

3.1.3 Etape 3

Dans le formulaire de sélection d'un praticien, remplacez le code de création de la liste déroulante par l'appel adéquat des fonctions que vous venez de développer. Testez votre code.

3.2 Réaliser la fiche du praticien

Lorsque l'utilisateur valide le choix d'un praticien en cliquant sur Ok, le formulaire est soumis ; les informations sur le praticien sont récupérées et la fiche du praticien est créée et affichée:

Concrêtement cette fiche est constituée du formulaire nommé `formPraticien`. Commencez par lui ajouter un attribut `name` avec la même valeur que l'`id`.

Ce formulaire ne sert ici que pour l'affichage des données. Ouvrez le fichier `cherchePraticien.php`. On pourrait en réutiliser le code, mais ce code n'est pas "propre". En particulier :
- à `select *` il faut préférer `select` suivis de la liste des colonnes souhaitées. Parce que :
  - le code est plus lisible,
  - le SGBD n'a pas à interpréter le *,
  - il n'est pas nécessaire de récupérer les colonnes que l'on n'utilise pas, comme ici `PRA_NUM`.
  - si plus tard on complète la structure de la table avec de nouvelles colonnes, l'exécution de la requête n'utilisera pas plus de ressources.

- Deux requêtes SQL sont exécutées, alors qu'une seule serait nécessaire.

- Le code HTML est mal conçu : la balise `<label>` est détournée de son usage pour structurer une zone d'affichage de données, présentée comme une zone de saisie de type `text` via une classe nommée "zone". Le comble étant que l'attribut `size` est utilisé pour spécifier la largeur de chaque zone … qui est ensuite mise d'office à 300 dans les attributs de style de la classe "zone" !!

Un formulaire destiné à de l'affichage simple peut être réalisé comme un formulaire classique dans lequel tous les contrôles de saisie on l'attribut *readonly* (`readonly="readonly"`).

La réalisation de ce formulaire va se faire selon la même démarche que pour la liste déroulante :

- mettez au point la requête SQL qui retourne les informations sur un praticien nécessaires à la fiche.
- réalisez une fonction nommée `getInfosPraticien()` qui prend en paramètre le numéro d'un praticien et retourne les informations pour la fiche praticien.
- Ajoutez à votre bibliothèque une fonction nommée `formInputText()` qui retourne le code HTML d'une zone de saisie de type texte, et qui prend en paramètre :
  - le label de la zone,
  - le nom de la zone,
  - son id,
  - sa valeur,
  - sa taille,
  - sa longueur maximale,
  - son index de tabulation,
  - un booléen indiquant si la zone est en lecture seule.

- Codez le formulaire. Les zones de saisie ont toutes la même taille (*size*) : 50.

Il faut ensuite compléter le code du premier formulaire pour que le bon programme PHP soit appelé lorsque le formulaire est soumis. Déterminez quel est ce programme et complétez le code du formulaire.

La fiche praticien doit s'afficher lorsque le premier formulaire est soumis, mais pas lors de la sélection de l'option "Praticiens" dans le menu général. Modifiez le code de votre page web pour respecter cette règle, sachant que la fonction `isset()` permet de tester l'existence d'une variable ...

3.3 Travailler les finitions

3.3.1 Au rechargement, le praticien choisi doit rester sélectionné dans la liste.

Testez le fonctionnement. Vous avez pu constater qu'au rechargement de la page, le praticien choisi est perdu ce qui est un comportement source d'erreurs pour l'utilisateur.

Dans une liste déroulante, il est possible de sélectionner par défaut une option de la liste en lui donnant l'attribut *selected* (`selected = "selected"`).

Pour savoir si un praticien a été choisi, il faut rechercher si un numéro de praticien a été transmis, dès le début du code en PHP :

```
if(isset($_REQUEST["lstPrat"]))
```

Proposez une solution à votre chef de projet pour qu'au rechargement de la page l'option par défaut de la liste soit la dernière sélectionnée. Une fois qu'elle sera validée, développez cette solution.

3.3.2 Tenir à jour la feuille de style

La présentation de la copie d'écran ci-dessus doit être réalisée. Il faut modifier la feuille de style. Pour éviter d'écrire sur les bords de la division "bas" :
- dans la division "bas" mettre un *padding-left* ou marge intérieure gauche et un *padding-bottom* ou marge intérieure inférieure de 20px.
- Dans la fiche praticien mettre un *padding-left* de 20px.
- Précisez que les labels de la fiche praticien s'affichent sous forme de bloc en ligne (*inline-block*) et qu'ils ont une largeur de 20% (de leur parent, c'est-à-dire du formulaire).
- précisez que les zones de saisie de la fiche ont une marge supérieure de 10px.

3.3.3 La validation par le W3C

Le W3C ne valide ni le JavaScript, ni le PHP. Ne pas soumettre une page en PHP. La méthode est de lancer la page, de choisir un praticien pour faire afficher ses renseignements. À ce stade, **tout le PHP a été exécuté par le serveur et a donc disparu** de la source.

Faire afficher la source de la page, la copier et la coller dans le *Validate by Direct Input* du http://validator.w3.org

- Il faut ensuite corriger les erreurs dans la page PHP d'origine. Si quantité d'erreurs html sont trouvées, copier aussi la source dans un nouveau document Notepad++, afin de disposer des numéros de ligne, qui ne sont pas les mêmes que dans la page PHP.
- Ne pas tenir compte des « erreurs » de JavaScript. Par exemple, il faut des guillemets autour des chaînes de caractères dans le JavaScript. Ces guillemets sont dénoncés comme inutiles, il faut pourtant les garder.

Valider gsb.css. La validation se passe à la page http://jigsaw.w3.org/css-validator/
Procédez par Saisie Directe, le fonctionnement est identique.