

# **Institut Supérieur des Etudes Technologiques en Communications de Tunis**

## **Projet de fin d'études**

**Implémentation d'une interface base de données web**

**Réalisé par**

Hamdi Mohamed Ali

Jabrane Khalil

TS5 – Réseaux et Informatique

**Encadré par**

Mr. Mejri Haikel

---

**2001-2002**

## *Dédicace*

*En témoignage de ma gratitude  
Et mon amour  
Je dédie ce travail  
A mes parents  
à qui je dois tout et dont les encouragements  
ont été pour moi le meilleur  
gage de réussite  
A toute ma famille  
A tous mes ami(e)s  
et à tous ceux qui me sont chers  
Veillez trouver ici le témoignage  
De mon amour et attachement .*

*Mohammed Ali* 

## *Dédicace*

*A ma chère mère*

*A mon brave père*

*A ma sœur*

*A mes frères*

*A toute ma famille*

*A tous mes fidèles amis*

*Khalil Jabrane...✍*

## *Remerciement*

Au terme de ce travail réalisé à l'Iset'Com, nous tenons vivement à remercier notre encadreur : **Mr. Mejri Haikel**, nous sommes particulièrement touché par son dévouement, sa compétence et sa disponibilité.

Nos profonds remerciements pour les membres du jury pour l'honneur qu'ils nous font en jugeant ce modeste travail.

Nos vifs remerciements s'adressent aussi à tous nos enseignants et tous le personnel de l'Iset'Com, pour leurs gentillesse et leurs encouragements .

A tous nos ami(e)s pour leurs aides précieuses ainsi que leurs convivialité tout au long de ce projet.

Veuillez accepter l'expression de nos remerciements les plus sincères.

***Hamdi & Jobrane .*** 

# Sommaire

<b>Introduction générale.....</b>	<b>1</b>
-----------------------------------	----------

## **Partie 1 - Etudes préliminaires**

<b>Chapitre I - Etude de la problématique.....</b>	<b>4</b>
I.1. Contexte générale.....	4
I.2. Notions des pages statiques et pages dynamiques.....	5
<b>Chapitre II - Etude des différentes techniques d'interfaçage               bases de données web.....</b>	<b>6</b>
II.1. Introduction.....	6
II.2. Technique d'interfaçage CGI(Common Gateway Interface).....	7
II.2.1. Notions HTML-CGI.....	7
II.2.2. Programme CGI.....	8
II.2.3. Pages dynamiques avec CGI.....	9
II.3. Technique d'interfaçage ASP(Active Server Pages).....	9
II.3.1. Les fonctions d'ASP.....	10
II.3.2. Fonctionnement de l'ASP.....	10
II.3.3. Le composant ADO(Active Data Object).....	11
II.3.4. Le modèle objet d'ASP.....	13
II.4. Technique d'interfaçage PHP(Professional HTML Preprocessing).....	15
II.4.1. Généralités.....	15
II.4.2. Fonctionnement et syntaxe.....	16
II.4.2.a. Syntaxe.....	16
II.4.2.b.Types de données.....	16
II.4.2.c. Opérations.....	16
II.4.3. Structures du langage PHP.....	17
II.4.3.a. Structures de test.....	17
II.4.3.b. Structures de contrôle.....	18
II.4.4. Les fonctions en PHP.....	19
II.4.4.a. Déclaration d'une fonction.....	19
II.4.4.b. Appel d'une fonction.....	20
II.4.4.c. Fonctions de base.....	20
II.4.4.d. Interaction avec les formulaires.....	21
II.5. Critères du choix de la technique PHP.....	22

## **Partie 2 - Présentation de l'environnement de travail**

<b>Chapitre I - Présentation générale de Linux.....</b>	<b>25</b>
I.1. Introduction.....	25

I.2. Hiérarchie des fichiers sous Linux.....	25
I.3. Installation de Linux.....	26
I.3.1. Préparatifs de l'installation.....	26
I.3.2. L'installation pas à pas.....	27
I.3.2.a. Disk Druid ou F.disk.....	27
I.3.2.b. Création de zone de Swap.....	28
I.3.2.c. Création du partition Linux.....	28
I.3.2.d. Formatage des partitions créées.....	29
I.3.2.e. Les composants Linux.....	29
I.3.2.f. Configuration finale.....	29
I.3.2.g. Installation de LILO(Last In Last Out).....	30
I.4. Utilisation du système Linux.....	30
<b>Chapitre II - Installation et configuration du serveur Apache.....</b>	<b>33</b>
II.1. Introduction.....	33
II.2. Fonctionnement d'Apache.....	33
II.3. Installation du serveur Apache.....	34
II.3.1. Installation à partir des packages.....	34
II.3.2. Installation à partir des sources.....	34
II.3.3. Arborescence des répertoires.....	35
II.4. Configuration d' Apache.....	35
II.5. Installation de PHP.....	36
<b>Chapitre III - Le SGBD MYSQL.....</b>	<b>38</b>
III.1. Définition de MYSQL.....	38
III.2. Organisation de données sous MYSQL.....	38
III.3. Installation de MYSQL.....	40
III.3.1. Installation à partir de la version binaire(rpm).....	40
III.3.2. Installation avec la version source rpm.....	40
III.3.3. Installation à partir de la distribution source.....	41

## **Partie 3 - Implémentation de l'interface base de données web**

<b>Chapitre I - Conception et implémentation de la base de données                   du bibliothèque.....</b>	<b>44</b>
I.1. Introduction.....	44
I.2. Objectif.....	44
I.3. Apport de la bibliothèque virtuelle.....	45
I.4. Conception de la base de données.....	45
I.4.1. Modélisation des données.....	45
I.4.2. Modèle Conceptuel de Données (MCD).....	46
I.4.3. Modèle Logique de Données(MLD).....	47
I.5. Implémentation de la base de données du bibliothèque.....	48
I.5.1. Création de la base de données.....	48
I.5.2. Création des tables dans la base.....	48
I.5.3. Insertion des tuples dans les tables créées.....	50

<b>Chapitre II : Réalisation de l'interface base de données web.....</b>	<b>51</b>
II.1. Introduction.....	51
II.2. La démarche de réalisation d'un site web.....	51
II.3. Arborescence générale du site.....	52
II.4. Organisation de la page d'accueil.....	52
II.5. Présentation de la page d'accueil.....	53
II.6. Procédure de recherche.....	54
II.6.1. Comment se fait la recherche.....	54
II.6.1. Comment se fait la recherche.....	54
II.7. Exemple de recherche de livre.....	55
II.7.1. Recherche de livre par titre.....	55
II.7.2. Résultat de la recherche.....	56
II.7.3. Exemple de script PHP.....	57
 <b>Conclusion Générale.....</b>	 <b>59</b>
<b>Glossaire.....</b>	<b>60</b>
<b>Bibliographie.....</b>	<b>63</b>

## **Liste des figures**

- Figure 1 : Pages dynamiques avec CGI
- Figure 2 : Fonctionnement d'ASP
- Figure 3 : Accès aux bases de données via ODBC
- Figure 4 : Structure générale du modèle objet d'ASP
- Figure 5 : Données sous forme d'enregistrement
- Figure 6 : Enregistrement sous forme de table
- Figure 7 : Tables sous forme de base de données
- Figure 8 : Base de données sous forme de SGBD
- Figure 9 : Modèle Conceptuel de Données
- Figure 10 : Modèle Logique de Données
- Figure 11 : Architecture du site
- Figure 12 : Organisation de la page d'accueil
- Figure 13 : Page d'accueil
- Figure 14 : Recherche par titre de livre
- Figure 15 : Résultat de la recherche



---

## Introduction Générale

---

Le web est le service le plus important de l'Internet. C'est un ensemble de documents sous forme électronique disposés sur des serveurs connectés à Internet.

Le web est une sorte d'immense document planétaire qu'on peut consulter et afficher le contenu. Compte tenu de sa facilité d'accès à l'information et ses différents objectifs d'utilisation (informer, publier, vendre, éduquer, publicité,...), il est devenu le moyen de communication numéro un du monde.

Ce service a subi une grande évolutivité ce qui a permis de passer de la notion des pages web statiques vers les pages web dynamiques.

Les difficultés rencontrées pour mettre à jour et assurer l'évolution des pages web statiques légitiment une orientation vers des solutions dynamiques. Ces pages HTML dynamiques seront produites, à partir des structures préétablies alimentées en informations, par des bases de données.

Une question se pose ici : c'est comment mettre en œuvre ses pages dynamiques ? Ou d'autre façon comment faire des interactions entre un tel site Web et une source d'informations comme les bases de données.

La solution c'est d'implémenter un intermédiaire entre une page HTML et une base de données. Cet intermédiaire peut être réalisé selon plusieurs techniques et assurant l'interfaçage entre le Web et la base de données présentant chacune ses propres méthodes et caractéristiques.

A la lumière de ce qui précède, notre projet consiste à élaborer en place une interface base de données web. Pour cela, le présent projet est subdivisé en trois parties essentielles :

- Une première partie qui sera consacrée à la présentation de la problématique accompagnée d'une étude des différentes techniques d'interfaçage bases de données et Web.

- Une deuxième partie qui présentera les différentes composantes constituant l'environnement de l'implémentation de notre interface base de données web.
- Une troisième et dernière partie qui sera consacré à la conception et l'implémentation de la base de données de la bibliothèque suivie de la réalisation de l'interfaçage du site web avec cette base.

## **Partie 1: Etudes préliminaires**

**Chapitre I : Etude de la problématique**

**Chapitre II : Etude des différentes techniques d'interfaçage  
bases de données web**

## Chapitre I

---

### Etude de la problématique

---

#### I.1. Contexte Général

Il s'agit de mettre en place une interface web et bases de données ou plus précisément, une passerelle applicative entre un serveur http (serveur web) et un système de gestion de bases de données. En effet, l'étude des paramètres d'implémentation d'une telle interface tels que client/serveur protocoles de communication, systèmes d'exploitations, langages de programmations est indispensable.

L'objectif visé tient à : Associer les principes de navigation hyper textuelle à la gestion de données structurées dans le but de produire de l'information dynamique générée et formatée "à la volée" en HTML en fonction des demandes exprimées par l'utilisateur.

Lors du développement d'une application qui accède à des bases de données via l'Internet, il faut envisager les situations suivantes :

- Présentation du résultat d'une requête de l'utilisateur.
- Mise à jour simple de la base de données par l'utilisateur.
- Possibilité de grouper plusieurs mises à jour dans une transaction.

Le premier cas permet la consultation des bases de données sans laisser la possibilité à l'utilisateur de modifier les données. Les deux autres autorisent la modification de la base par l'utilisateur, avec ou sans système transactionnel.

Au delà de la présentation des techniques d'interfaçage entre bases de données et web et dans le but de proposer de l'information dynamique, il est important de souligner que l'argument de poids, qui a conduit au choix de cette solution réside dans sa principale performance : la rapidité.

Malgré que la complexité du contexte de mise en place d'un site web dynamique conduit à une diversité des solutions disponibles.

## **I.2. Notions des pages statiques et pages dynamiques**

La requête de l'utilisateur, qui s'exprime à travers un lien activé ou un formulaire dûment rempli, ramène systématiquement de l'information au format HTML

La vision statique ou dynamique de la page HTML est bien entendu une vision « coté serveur » c'est à ce niveau que deux processus distincts vont être mis en œuvre en fonction du type d'opération demandée (lien vers une page ou un script associé à un formulaire ).

Les pages statiques sont des pages HTML invariables préparées à l'avance. Le serveur web renvoi ces pages à l'utilisateur mais n'effectue aucune action particulière.

Les pages dynamiques, quant à elles, sont créées en réponse à une demande d'un utilisateur. Lorsque l'utilisateur clique sur un bouton d'un formulaire, les données du formulaire sont envoyées au serveur web celui-ci réagit en transmettant les données à un script ou une application en vue de les traiter en interrogeant une base de données ou en y intervenant, le serveur renvoi ensuite les résultats à l'utilisateur dans une page HTML.

Pour générer des pages dynamiques et rendre un site web plus dynamique et agréable et/ou utile, les techniques d'interfaçages tel que ASP, CGI et PHP, vont être analysés afin d'explorer les caractéristiques générales de chaque technique ainsi que ses particularités.

## Chapitre II

---

### **Etude des différentes techniques d'interfaçage bases de données Web**

---

#### **II.1. Introduction**

Plusieurs services "on line" permettent de générer des pages webs dynamiques en se connectant à une base de données afin de permettre une consultation des données qui y sont stockées.

Bien évident, pour développer de tels services, il suffit de relier le serveur web à la base de données de sorte qu'il renvoie le contenu de la base sur demande.

En effet, pour mettre en évidence un tel lien, on a recours aux techniques d'interfaçages entre bases de données et web tels que :

- CGI : Common Gateway Interface
- ASP : Active Server Pages
- PHP : Professional HTML Preprocessing

## II.2. Technique d'interfaçage CGI (Common Gateway Interface)

Ce n'est pas chose récente que de créer des pages webs dynamiques en s'appuyant sur des bases de données et des applications s'exécutant sur le serveur. Traditionnellement, ces pages utilisent l'interface CGI. Ce mécanisme permet à des clients web d'exécuter des programmes sur un serveur web, qui de sa part renvoie le résultat au client à travers l'utilisation des programmes CGI, le serveur reçoit les entrées d'une page HTML et produit dynamiquement des documents HTML en retour.

CGI est en fait un standard multi-plateforme et multi-applications. Tous les serveurs webs devraient mettre en œuvre CGI de la même façon, de manière à ce qu'une application CGI puisse fonctionner sur tous systèmes.

Les applications CGI sont portables sur différents serveurs web au niveau applicatif, c'est leur principale attrait. De plus, le code source sur la plupart des applications courantes est disponible gratuitement.

### II.2.1. Notion HTML - CGI

L'interaction entre HTML et un programme CGI se fait à travers des « *tags form* » (formulaires) HTML. Il peut être appelé par un simple lien hypertexte utilisant le « *tag ancre* » (<a>) comme ce qui suit : < a href = « *programme CGI* »> .

Le programme CGI qui figure dans ce lien hypertexte sera appelé d'une certaine façon (ou activité) comme s'il l'avait été par un formulaire HTML. Le formulaire HTML permet d'envoyer des données dynamiques au programme CGI et ces données sont transmises au serveur pour être utilisées par le programme CGI, mais la façon de construction du formulaire HTML définit la façon dont les données seront envoyées. Le browser, par la suite, se charge de la majeure partie de formatage des données HTML avec ses « *tags form* », et permet la collecte et l'envoi des données vers le programme CGI correspondant. Cet envoi des données se fait grâce à l'entête de requête du protocole de transport hypertexte (http).

Les tags HTML indiquent au browser quel type d'entête http est nécessaire pour communiquer avec le serveur et avec le programme CGI en particulier. Les entêtes http de bases qui permettent de communiquer avec un programme CGI sont « *GET* » et « *POST* » (valeur de l'attribution « *method* » de l'entête http) :

Si la méthode « *GET* » est utilisée alors les données seront envoyées via le champ URI (Identificateur Unifié de Ressource).

Si c'est la méthode « *POST* » qui est utilisée, les données seront envoyées sous forme d'un message séparé, après que tous les en-têtes http soient été envoyés.

Le serveur reçoit l'en-tête http et appelle le programme CGI dont l'emplacement est défini par cette entête envoyé. Le serveur collabore avec le programme CGI activé pour renvoyer au browser une réponse http.

La méthode la plus usuelle pour renvoyer des réponses http consiste cependant pour le programme CGI à générer le minimum d'en-tête http nécessaires. Un en-tête réponse http « *content type* » suffit généralement.

### II.2.2. Programme CGI

Le programme CGI doit surveiller les en-têtes des requêtes http qui arrivent et doit aussi renvoyer des en-têtes réponses http. La programmation CGI consiste à écrire des applications tenant lieu d'interface ou de programme de passerelle entre le browser client, le serveur web et une application classique (base de données comme exemple).

Le principe de programmation CGI consiste à construire le document HTML correspondant à un lien hypertexte au moment même où l'on clique sur ce lien. Ce document sera envoyé au client au fur et à mesure de sa construction. Les programmes lancés à partir de liens exécutables sont appelés des scripts CGI.



### II.2.3. Pages dynamiques avec CGI

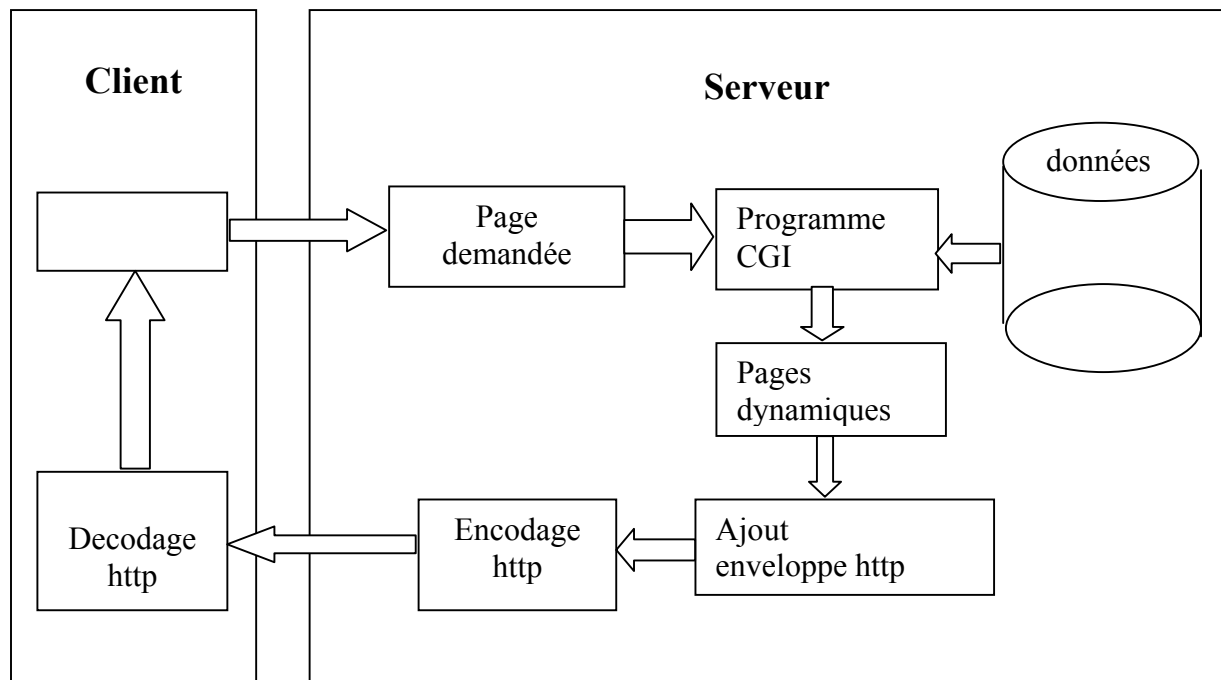


Figure 1 : Pages dynamiques avec CGI

### II.3. Technique d'Interfaçage ASP(Active Server Page)

L'ASP (Active Server Pages) fournit un cadre pour les applications côté serveur et facilite considérablement la construction d'applications web, pages web dynamiques. Le document ASP peut contenir à la fois du code HTML et des scripts exécutables sur le serveur et permettant aux développeurs de site web et aux fournisseurs d'accès Internet de rendre leurs pages encore plus agréables, dynamiques et interactives.

ASP offre la souplesse puisqu'il peut agir avec les techniques de web dynamiques existants tel que CGI.

Une page ASP peut être écrite dans n'importe quel langage de script active X. L'écriture d'une page ASP se résume à la rédaction d'un code côté serveur qui produit le résultat que l'on souhaite.

### II.3.1. Les fonctions d'ASP

Parmi les fonctions existantes de l'ASP il y a :

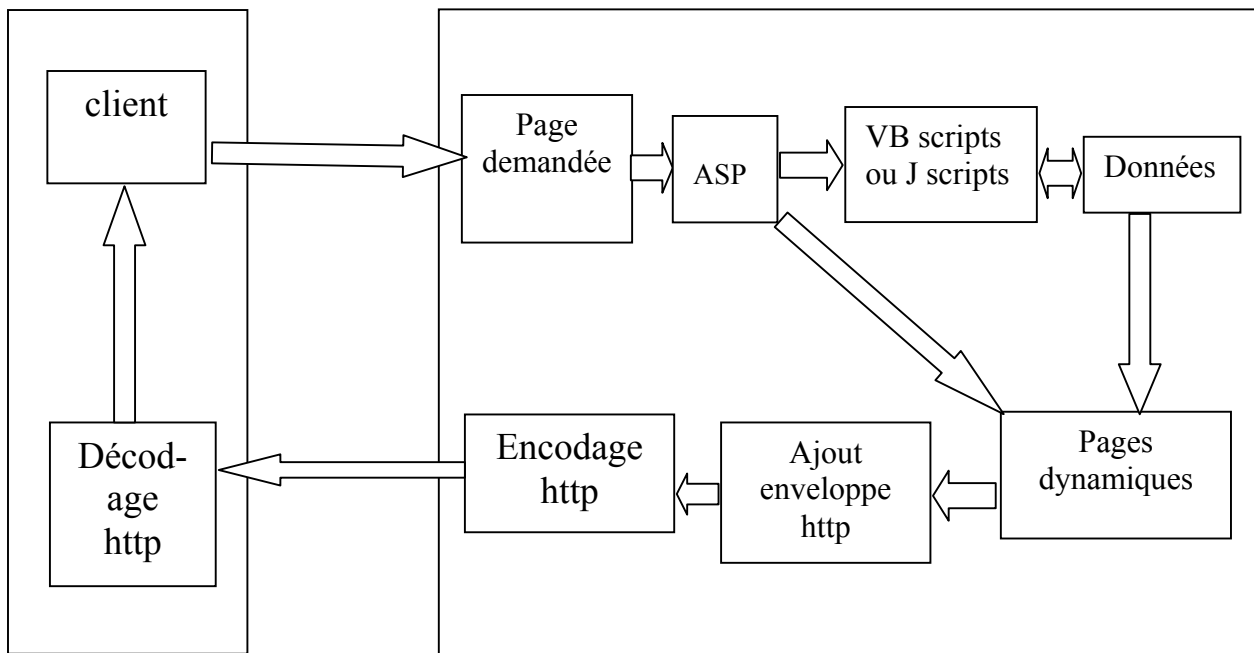
- *Les scripts côté serveur* : ASP traite des scripts qui peuvent être incorporés dans des pages HTML. Ces scripts seront exécutés sur le serveur Web et non par le navigateur.
- *La possibilité de consulter les bases de données Web* : Les scripts côté serveur permettent de créer facilement de puissantes fonctions de gestion de base de données Web.
- *Les Active Server Components* : Permettent d'utiliser des composants programmables à l'aide du Visual Basic, Visual C++, etc. Permet d'intégrer des applications Web dans des systèmes client-serveur existants.

### II.3.2. Fonctionnement de l'ASP

Les fichiers ASP permettent de générer dynamiquement des balises HTML dans le code source de la page web ASP avant son envoi au navigateur.

Avant de la renvoyer au navigateur, le serveur vérifie l'existence d'un ou plusieurs scripts à gérer et à exécuter. Ces scripts peuvent calculer et insérer des valeurs dans la page ou créer du texte ou du code HTML supplémentaire en fonction du contexte. A la fin, le serveur la place dans l'enveloppe http et l'envoie au navigateur.

Quand elle arrive à la destination, elle ne contient que du texte et du code HTML, tout comme une page statique habituelle. Dans ce cas le navigateur pointe réellement sur le fichier envoyé. Cependant, c'est une copie du fichier, modifiée d'après le résultat de l'exécution des scripts que l'utilisateur visualise.



**Figure 2 : Fonctionnement d'ASP**

Une page dynamique peut donc contenir des scripts renvoyés au navigateur, ainsi que des scripts exécutés sur le serveur.

Le serveur doit pouvoir distinguer un script à exécuter pour créer dynamiquement la page, d'un script à envoyer tel qu'elle au navigateur.

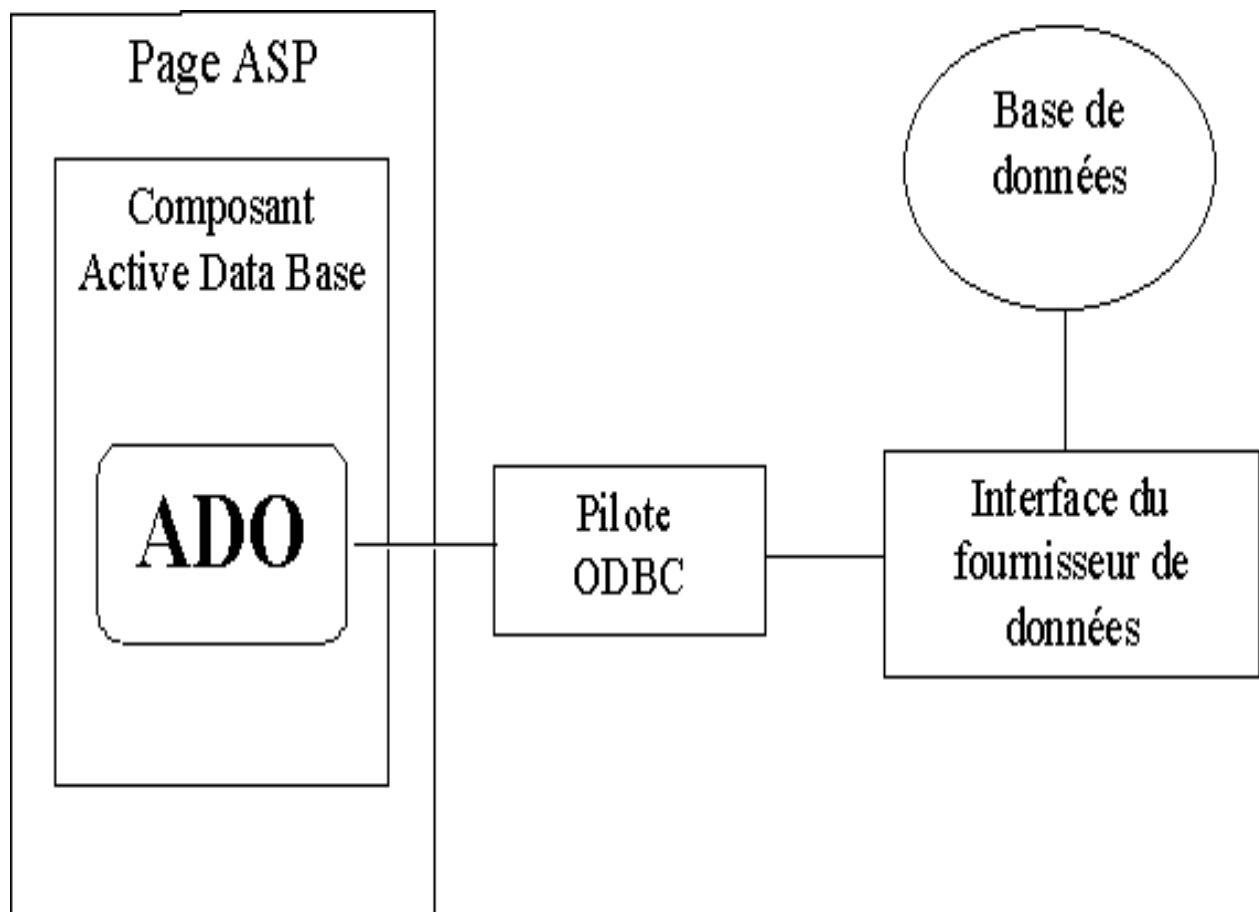
ASP utilise les balises `< ./. ..... ./>` pour baliser des scripts devant être exécutés sur le serveur, alors que le navigateur attend des scripts compris entre les balises HTML `<script>` et `</script>`. Une page ASP peut inter-agir avec toutes les techniques web dynamiques existantes ce qui donne l'avantage de compatibilité.

### II.3.3. Le composant ADO (Active Data Object)

L'intérêt principal des pages dynamiques pour un site web est de fournir un accès aux bases de données. Un des composants livrés en standard avec ASP est conçu pour répondre à ce besoin.

Il s'agit du composant ADO qui permet d'accéder à n'importe quel SGBD pour qu'il existe un pilote ODBC (Open Data Base Connectivity). On peut, avec ODBC, n'utiliser que ce composant (ADO) pour accéder aux différents types de base de données, ce qui simplifie le processus de création et de maintenance du site.

Cependant, ADO est construit sur une autre couche appelée (OLE –DB) qui met en relief l'interface de données appropriées basée sur l'ensemble de méthodes et propriétés.



**Figure 3 : Accès aux bases de données via ODBC**

ADO est un mécanisme simple et clair pour accéder aux bases de données. La hiérarchie ADO comporte trois objets principaux :

**\*Objet connexion :**

Sert à établir une connexion active qui permet l'accès aux données, stockées dans une base de données.

**\*Objet command :**

Sert à récupérer des enregistrements à partir de la source de données et à exécuter des requêtes SQL ou manipuler les données.

**\*Objet recordset :**

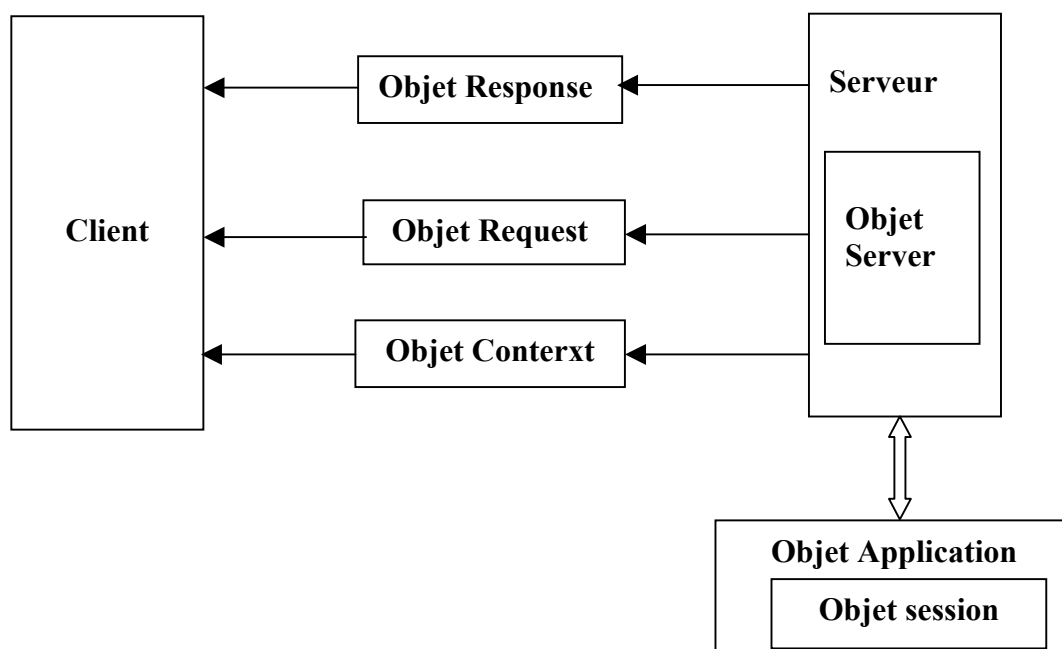
Donne accès aux données envoyées par une requête SQL.

### **II.3.4. Le modèle objet d'ASP**

Bien que l'on puisse insérer des scripts dans une page ASP, il reste toujours des contraintes dues aux limites intrinsèques des langages de scripts. Seulement ces contraintes possèdent une qualité majeure, du moment qu'ils permettent d'agir sur les objets appartenant à l'environnement d'ASP, via leurs méthodes et propriétés. Autre point important à signaler est que ASP propose un ensemble d'objets manipulables par scripts.

**\*Structure générale du modèle objet d'ASP**

ASP, possède cinq objets prédéfinis utilisables dans les échanges client/serveur et organisés hiérarchiquement comme suit.



**Figure 4 : Structure générale du modèle objet d'ASP**

**\*Objet application et session :**

Permettent de suivre les différentes activités d'un utilisateur et de proposer les réponses relatives à chacun d'eux. Les concepts d'application et de session sont au cœur même de la programmation ASP.

**\*Objet request :**

Contient les informations concernant la requête de l'utilisateur.

**\*Objet réponse :**

Il s'occupe de ce qui arrive au serveur. L'objet-réponse consiste à traiter les informations qui vont être renvoyées au client.

**\*L'objet server :**

Il est doté des méthodes et de propriétés utilisées dans les pages ASP.

**\*L'objet Objet Context :**

Cet objet est destiné à traiter ou à terminer des transactions initiées par un script ASP.

## **II.4.Technique d'interfaçage PHP (Professionnal HTML Preprocessing)**

### **II.4.1 Généralités**

PHP est un langage de programmation permettant de générer des pages HTML de manière dynamique. C'est une solution professionnelle permettant de séparer le fond et la forme des pages web. On peut aussi aller chercher les informations à mettre à la disposition des clients consultant un site dans une base de données ou encore gérer des sections confidentielles du site, avec des accès réservés, authentifiés par mot de passe.

Tout simplement, PHP est largement inspirés des langages C et Perl. Pour utiliser la technologie PHP, il faut disposer d'un serveur web avec les extensions (modules) PHP y sont installées. Tout code PHP est inclus dans un fichier HTML, l'extension du fichier contenant ce code est déterminé par le serveur web.

Il faut indiquer le passage en code PHP par la balise d'ouverture du code « `< ? PHP` », puis lorsque le code est terminé ou pour revenir à une écriture HTML, on écrira la balise de fermeture « `?>` ». On peut inclure dans un fichier autant de sections PHP que l'on veut, et mélanger ainsi les langages HTML et PHP à volonté.

Comme beaucoup de langages, chaque instruction doit se terminer par un caractère spécial, ici par le point virgule ( ; ).

## II.4.2. Fonctionnement et syntaxe

En fonction des serveurs web, PHP peut être exploité en tant qu'extension du logiciel http ou bien en tant que programme CGI indépendant.

### II.4.2.a. Syntaxe

La syntaxe du langage PHP est inspirée des langages C et Perl. Il est à noter que tout nom de variable commence par le signe dollar ( \$ ), quel que soit le type de la variable, et qu'il est suivi par des caractères alphanumériques. Pour le reste de la syntaxe, elle est fortement semblable de celle du C.

### II.4.2.b. Type de données

-*Type scalaire* : il s'agit des entiers et réels.

-*Type tableau et chaîne de caractères* : Ils sont gérés de la même manière. Les chaînes de caractères étant en fait des tableaux de caractères, l'élément numéro « *i* » est dénoté par *\$tab[\$i]*.

### II.4.2.c. Opérations

-*Opérateurs logiques et de comparaison* : permettent d'effectuer les tests dits booléens, c'est à dire n'ayant que deux valeurs possibles vrai ou faux.

**Exemple** : *And, Or, XOR, <, >, = ...*

-*Opérateurs arithmétiques* : + , \* , - , ./.

-*Opérateurs d'affectation* : lorsqu'on affecte une valeur à une variable.

**Exemple** : *\$a = \$a + 2* ou *\$a += 2*.



### II.4.3. Structures du langage PHP

#### II.4.3.a. Structures de test

Ces structures permettent de choisir une action en fonction de conditions appelées aussi : « *structures alternatives* », tels que :

➤ **If**

If (condition)

    Instruction;

end if

If (condition)

    Instruction 1;

else

    Instruction 2;

end if

➤ **Switch**

Sert pour tester la même variable selon plusieurs valeurs.

-Switch (variable) {

    case 0: instruction 1;

break;

    case1: instruction 2;

break;

```
        case 2: instruction 3;

break;

        default: instruction 4;

break ;

} ;
```

#### II.4.3.b. Structures de contrôle

Elles permettent de contrôler l'ordre d'exécution d'un programme en effectuant par exemple des boucles :

➤ **boucle générique :**

While (condition)

Instruction 1;

Instruction 2;

End while.

➤ **Cas particulier : répéter**

Do {

<Instruction 1>;

<Instruction 2>;

} While (condition);

➤ **Compteur for**

```
For (i = 1; i = b; i ++){  
  
    <Instruction 1> ;  
  
    <Instruction 2> ;  
  
} ;
```

## **II.4.4 Les fonctions en PHP**

On appelle fonction un sous-programme qui permet d'effectuer un ensemble d'instruction par simple appel de la fonction dans le corps du programme principal.

Les fonctions permettent d'exécuter, dans plusieurs parties du programme, une série d'instructions ; cela permet une simplicité du code et donc une taille de programme minimale. D'autre part, une fonction peut faire appel à elle-même, on parle alors de fonction récursive

(il ne faut pas oublier de mettre une condition de sortie au risque de ne pas pouvoir arrêter le programme...).

### **II.4.4.a. Déclaration d'une fonction**

PHP recèle de nombreuses fonctions intégrées permettant d'effectuer des actions courantes. Toutefois, il est possible de définir des fonctions, dites « fonctions utilisateurs » afin de simplifier l'exécution de séries d'instructions répétitives.

Contrairement à de nombreux autres langages, PHP nécessite que l'on définisse une fonction avant que celle-ci ne soit utilisée, car pour l'appeler dans le corps du programme il faut que l'interpréteur la connaisse, c'est à dire qu'il connaisse son nom, ses arguments et les instructions qu'elle contient. La définition d'une fonction s'appelle « déclaration » et peut se faire n'importe où dans le code.

La déclaration d'une fonction se fait grâce au mot clé "*function*" selon la syntaxe suivante :

*Function* ( nom de la fonction) (type 1 *argument 1*, type 2 *argument 2* ...)

{

*<instruction 1>*

*<instruction 2>*

...

*<instruction n>*

}

#### II.4.4.b. Appel d'une fonction

Pour exécuter une fonction, il suffit de faire appel en écrivant son nom suivi d'une parenthèse ouverte puis d'une parenthèse fermée : « *Nom de la fonction ( ) ;* »

#### II.4.4.c. Fonctions de base

PHP fournit un grand choix de fonctions permettant de manipuler les bases de données. Toutefois, parmi celles-ci on distingue quatre fonctions essentielles :

- la fonction de connexion au serveur.
- la fonction de choix de la base de données.
- la fonction de requête.
- la fonction de déconnexion.

Avec le système de gestion de bases de données (SGBD) MySQL les fonctions PHP sont :

- ***mysql\_affected\_rows*** : donne le nombre de lignes affectées par les commandes **INSERT, UPDATE, DELETE...** ;

- ***mysql\_close ( )*** : fermeture de session, mysql.
- ***mysql\_connect\_db ( )*** : ouverture de session mysql.
- ***mysql\_create\_db ( )*** : créer une base de donnée mysql.-mysql\_db\_name : donne le nom de base en cours de consultation.
- ***mysql\_db\_query ( )*** : envoie une requête SQL sur une base de donnée précise en paramètre.
- ***mysql\_fetch\_array ( )*** : Ramène un tableau correspondant aux lignes effectuées par la dernière requête **SELECT**.
- ***mysql\_field\_name ( )*** : donne le nom d'un champ.
- ***mysql\_select\_db ( )*** : sélectionne une base de donnée.
- ***mysql\_table\_name ( )*** : donne le nom de la table d'un champ.
- ***mysql\_num\_rows*** : Permet de connaître le nombre de lignes retournées par des requêtes.
- ***mysql\_num\_fields*** : Permet de connaître le nombre de champ présent dans un tuple résultant d'une requête.
- ***mysql\_list\_tables*** : Permet d'obtenir la liste des tables d'une base de données précise (la dernière à la quelle on a été connectée).

#### II.4.4.d. Interaction avec les formulaires

Les modes de communication entre pages PHP et pages HTML, reposent comme tous les mécanismes d'interfaçage, sur les liens hypertextes « *<A HREF.... > et </A>* » et les formulaires « *<FROM> et </FORM>* ».

➤ **utilisation des liens :**

Les liens HTML exprimés au moyen des balises <A> et <\A> peuvent être associés à :

- page HTML.
- une image.
- un lien destiné au navigateur.

Et autres.....

➤ **utilisation des formulaires :**

Pour réaliser le traitement du contenu d'un formulaire par une page PHP, il suffit d'associer à l'attribut « Action » de la balise <FORM> et <\FORM>, le nom de la page PHP.

## **V. Critères de choix de la technique PHP**

Parmi les différentes techniques d'interfaçage entre Base de Données et web on a choisit pour notre projet la technique PHP. C'est un langage puissant qui s'apparente au C, et au CShell par sa syntaxe, et ses nombreuses fonctionnalités héritées du monde Unix qui explique la facilité d'implémentation.

Aussi PHP est délivré gratuitement avec la distribution Red Hat de Linux qui représente le système d'exploitation utilisé pour ce projet d'interface entre Base de données et Web.

Cet outil avec ses macro-fonctions présente des possibilités plus étendues que celles des ASP (Active Server Pages) et CGI (Common Gateway Interface). Avec ses fonctionnalités croissantes au fil des versions, PHP devient un véritable langage de script intégrable dans des pages HTML.

Les pages PHP fonctionnent sur de nombreuses plates-formes Unix (Solaris, Linux, OS/2...) et sur plate-forme Microsoft et facilitent l'accès à un grand nombre de bases de données : Oracle, Sybase, MySQL, Postgres SQL, Adabas, Informix, etc.

PHP, MySQL est incontestablement le couple le plus en vogue sur Internet. Ces deux oiseaux ont une complémentarité qui n'est plus à démontrer, le premier, un langage de script très puissant dans le cybermonde, le second, une base de données relationnelle très populaire.

De plus, PHP fonctionne, préférentiellement, avec le serveur Apache (en jouant le rôle d'un module intégré à Apache), comme il peut fonctionner en mode CGI avec tous les autres serveurs web ; et à chaque fois les scripts PHP ne sont pas pré-compilés mais "interprétés" par le serveur web.

## **Partie 2 : Présentation de l'environnement de travail**

**Chapitre I : Présentation générale de Linux**

**Chapitre II : Installation et configuration du serveur  
Apache**

**Chapitre III : Le SGBD MYSQL**



## Chapitre I

---

# Présentation générale de Linux

---

## I. 1. Introduction

Du point de vue du système d'exploitation, Linux est un système proche des systèmes Unix. Linux est un système **multi plate-forme**. Il est également **multi-utilisateur** (plusieurs personnes peuvent en même temps travailler sur le même ordinateur), mais aussi **multi-tâche** (plusieurs applications peuvent être lancées en même temps sans qu'aucune n'affecte les autres) et **multi-processeur**. C'est un système **fiable, robuste, puissant et efficace**. Il utilise très peu de ressource et fonctionne quasiment aussi bien sur un ordinateur bas de gamme que sur un haut de gamme. Il est **distribué gratuitement** avec son code source. Un des principaux avantages de Linux est de supporter en standard le protocole TCP/IP, une famille de protocoles utilisés sur Internet. **Linux est donc un moyen gratuit de créer un réseau local, de se connecter à Internet et de mettre en place un serveur.**

X-Window est l'interface graphique des stations UNIX. L'avantage majeur de ce type d'interface est l'utilisation d'une souris pour remplacer certaines commandes. Sous Linux il existe une implémentation du système X-Window appelée **Xfree86**.

## I.2. Hiérarchie des fichiers sous Linux

Pour assurer la compatibilité et la portabilité, les systèmes Linux respectent l'unique norme FHS (**File Hierarchy Standard**). La hiérarchie de base est la suivante:

- / : la racine, elle contient les répertoires principaux.

- **/Bin** : Contient les exécutables essentiels au système employé par tous les utilisateurs.
- **/Boot** : Contient les fichiers permettant à Linux de démarrer.
- **/Dev** : Contient les points d'entrées des périphériques.
- **/Etc** : contient les commandes et les fichiers nécessaires à l'administrateur système.
- **/Home** : répertoire personnel à l'utilisateur.
- **/Lib** : contient des bibliothèques partagées essentielles au système lors du démarrage.
- **/Mnt** : Contient le point de montage des partitions temporaires ( cd-rom, disquette..).
- **/Opt** : Contient des packages d'applications supplémentaires.
- **/Root** : Répertoire de l'administrateur root.
- **/Sbin** : Contient les binaires système essentiels (par exemple la commande **adduser**).
- **/Tmp** : Contient les fichiers temporaires.

La hiérarchie secondaire est la suivante :

- **/usr/bin** : Contient la majorité des fichiers binaires et commandes utilisateurs.
- **/usr/include** : Contient les fichiers d'en-tête pour les programmes C et C++.
- **/usr/lib** : Contient la plupart des bibliothèques partagées du système.
- **/usr/local** : Contient les données relatives aux programmes installés sur la machine locale par le root.
- **/usr/sbin** : Contient les fichiers binaires non essentiels au système réservés à l'administrateur système.
- **/usr/share** : Réservé aux données non dépendantes de l'architecture.
- **/usr/src** : Contient des fichiers de code source.
- **/var** : Contient des données variables.

## I.3. Installation de Linux

### I.3.1. Préparatifs de l'installation

Avant de lancer l'installation de Linux, le disque dur doit avoir deux partitions au minimum. La première contiendra l'arborescence du système d'exploitation et des programmes, tandis que la seconde constituera une zone tampon similaire à la mémoire vive.

Donc installer Linux en tant que système d'exploitation seul ou en combinaison avec d'autres, ne changera pas les partitions nécessaires pour Linux. On doit créer au moins les deux partitions mentionnées avant. Pour que Linux fonctionne correctement et d'une manière optimale. Créer une partition d'échange ( **swap** ) de taille double à celle de la mémoire vive constitue une bonne approximation.

### **I.3.2. L'installation pas à pas :**

Le programme d'installation considéré figure sur un CD-ROM, on procède au démarrage de l'ordinateur avec la disquette d'amorçage. La première étape consiste à définir les paramètres spécifiques de notre pays c'est à dire à réaliser l'installation en français en utilisant un clavier français. On précise ensuite qu'on souhaite réaliser l'installation à partir du CD-ROM. le programme d'installation a maintenant accès à tous les logiciels et à tous les composants figurant sur le CD. Après la confirmation des divers écrans d'accueil du programme d'installation de Red Hat, effectue les réglages spécifiques. Si on sélectionne l'icône installation complète du système Linux, il affiche ensuite trois classes d'installation :

- Installation relativement rapide et typique de type station de travail autonome(workstation) .
- Installation complète visant à tenir le rôle de serveur.
- Installation traditionnelle ( custom ).

#### **I.3.2.a. Disk Druid ou Fdisk**

L'utilitaire de la partition de la distribution Red Hat s'appelle Disk Druid, sa particularité est que l'utilisateur le commande par des menus. En outre, ce logiciel vérifiera si les réglages adoptés sont adaptés à une installation standard. Si on s'en écarte, les contrôles réalisés par Disk Druid empêchent la réalisation de la tâche. On doit dans ce cas utiliser l'utilitaire standard de Linux fdisk. Celui – ci autorise la libre définition des partitions, sans contrôle de sa part. Une fois les partitions sont créées, on appelle Disk Druid pour effectuer quelques réglages supplémentaires que fdisk ne réalise pas. En suivant Disk Druid, on doit créer au moins une partition de swap et une autre Linux.

### I.3.2.b. Création de zone de Swap

Pour créer la partition de Swap, on choisit le disque dur sur lequel elle sera créée en suivant les étapes suivantes :

- Activer la bouton '*Ajouter*' ( dans la fenêtre '*partitions actuelles du disque*' ) via la touche '*Tab*' , une fenêtre s'affiche alors dont le nom est '*Edit new partition*' .
- Passer la zone de saisie '*Mount point*' avec *Tab* aussi.
- Dans la zone size ( Mo ) on donne la taille en Méga-octet.
- Passer la zone '*Growable*' avec la touche *Tab*.
- On parvient par suite à la liste des '*types des partitions*', on sélectionne '*Linux Swap*' pour définir la zone de swap.
- Dans la liste '*Allowables Drives*' , on choisit le lecteur sur lequel sera créée cette partition.
- Enfin, pour quitter cette fenêtre et revenir au fenêtre précédente, on agit sur '*Ok*'.

### I.3.2.c. Création du partition Linux

Il est obligatoire de créer une partition système, laquelle recevra les logiciels du système d'exploitation. On se précède de la manière suivante :

- Choix du '*Ajouter*' , comme pour la zone de Swap .
- Indiquer le point de montage de la partition, / pour la partition système, on passe à la zone suivante par la touche '*Tab*'.
- On donne la taille exacte de la partition en *Mo*.
- La touche '*Tab*' nous fait ensuite passer à la liste des '*types des partitions*' dont on choisit '*Linux Native*' .
- Dans la zone '*Allowable Drives*' . on définit le disque dur sur lequel la partition sera créée.

- Enfin, on confirme par 'Ok '.

#### **I.3.2.d. Formatage des partitions créées**

Les étapes suivantes consistent à formater les partitions créées, c'est à dire que le système y créera une structure leur permettant de recevoir les données. Après qu'on quitte Disk Druid, une fenêtre s'affiche ' *Activation de l'espace de swap* ', on coche l'option '*vérification des mauvais blocs pendant le formatage*' , puis ' Ok '. On doit ensuite formater les partitions Linux déjà définies, une fenêtre '*partitions à formater* ' s'affiche.

#### **I.3.2.e. Les composants Linux**

L'étape suivante de l'installation est la sélection des composants logiciels ( Packages ) et ceux du système d'exploitation à copier du CD-ROM vers le disque dur tels que :

- RPM (Red Hat Packages Manager) .
- X Window système.
- Dos (Windows connectivity) .
- SMB (Samba) .
- DNS.
- C Developpement.

#### **I.3.2.f. Configuration finale**

XFree86 est l'interface graphique sous Linux (X-Window est l'interface des stations UNIX). Pour configurer XFree86, il suffit de taper "XF86Setup" en ligne de commande, en respectant les majuscules. Un écran apparaît vous demandant des informations sur votre écran et permettant de régler l'affichage. Une fenêtre graphique s'affiche alors, dans laquelle vous avez des onglets permettant de configurer chacun des éléments matériels de votre ordinateur. Il faudra donc configurer tout d'abord la souris, celle-ci étant la plupart du temps une souris de type Microsoft (ou PS/2 si elle est branchée sur un port spécial), sur le port COM1 (ou COM2).

Le clavier n'est généralement pas à modifier, il faudra veiller par contre à bien régler les paramètres de la carte vidéo car ceux ci se répercuteront directement sur l'affichage, causant un affichage peu stable en cas d'erreur (écran qui vacille par exemple).

Une fois tous les paramètres saisis vous quittez le programme de configuration et retournez en ligne de commande sous la console.

Il suffit de taper "**start x**" pour démarrer *x-window*.

### I.3.2.g. Installation de LILO (Last IN Last Out)

Dans la fenêtre ' Installation de LILO ' on choisit l'emplacement où LILO sera installé ( on peut choisir le ' Master Boot Record ' ).

Une fois LILO est copié dans le Master Boot Record, l'installation est achevée.

## I.4. Utilisation du système Linux

Les principales commandes de Linux se présentent dans le tableau ci-dessous :

Commande	Description	équivalent DOS
<b>ls</b>	liste le contenu d'un répertoire	dir
<b>cd</b>	change de répertoire	cd
<b>cd ..</b>	répertoire parent	cd
<b>mkdir</b>	crée un nouveau répertoire	md
<b>rmdir</b>	supprime un répertoire	deltree
<b>cp</b>	copie de fichier	copy, xcopy
<b>mv</b>	déplacement de fichier	move
<b>rm</b>	supprime le fichier	del
<b>passwd</b>	change le mot de passe de l'utilisateur	
<b>cat</b>	affiche le contenu du fichier	type
<b>more</b>	affiche le contenu du fichier avec des pauses	type  more

<b>man</b>	aide sur la commande demandée	help
<b>apropos</b>		
<b>lpr</b>	imprime le fichier demandé	print
<b>chmod</b>	<p>change l'attribut d'un fichier</p> <p><b><i>chmod XXX fichier</i></b></p> <p>XXX= Utilisateur Groupe Autres ou X représente un entier 1&lt;X&lt;7</p> <p>Lecture=1, Ecriture=2, Exécution=4</p> <p>X=Lecture+Ecriture+Exécution</p>	
<b>chfn</b>	change les informations personnelles vues avec finger	
<b>chsh</b>	Change le shell : <b><i>chsh user emplacement_du_shell</i></b>	
<b>finger</b>	liste des utilisateurs en ligne	
<b>traceroute</b>	trace le chemin entre la machine locale et la machine visée	
<b>ftp</b> <b>[machine]</b> <b>[port]</b>  <b>get</b>  <b>put</b>  <b>quit</b>	<p>transfert de fichier entre la machine locale et la machine cible</p> <p>récupère un fichier</p> <p>envoie un fichier</p> <p>quitte la session FTP</p>	
<b>telnet [machine]</b>	effectue un <u>telnet</u>	

<b>talk</b>	permet de parler à un utilisateur connecté  talk user	
<b>mesg</b>	autorise ou non la commande talk  <b><i>mesg n</i></b> : Empêche la réception de messages <b>talk</b>  <b><i>mesg y</i></b> : Permet la réception de messages <b>talk</b>	
<b>bye</b>	déconnexion	



## Chapitre II

---

# Installation et configuration du serveur Apache

---

### II.1. Introduction

Apache est un serveur Web libre, c'est le standard comme serveur Web sous linux, mais aussi le serveur Web le plus utilisé sur Internet avec plus de 60% des sites d'Internet. Donc cela ne peut que prouver sa stabilité et ses performances. C'est un serveur World Wide Web utilisant le protocole HyperText Transfert Protocol. Il tourne, généralement, sous un OS multitâche dont l'exécutable(binaire) se nomme httpd. Il est très souvent disponible en standard sur les distributions mais on doit souvent le configurer pour ses propres besoins.

### II.2. Fonctionnement d'Apache

Apache se présente sous la forme d'un fichier exécutable nommé httpd, qui, en terminologie Unix signifie : *Démo http*. Dans le monde Unix un démo est un programme redondant du système auquel on peut faire appel à n'importe quel moment. Ainsi, httpd est appelé à chaque fois qu'un client du réseau demande par exemple, l'affichage d'un document HTML.

Ce type d'exécutable tournant en arrière plan peut être comparé au serveur de Windows NT. Outre cet exécutable, le serveur apache puise les informations de sa configuration dans le répertoire *Conf* de son répertoire d'installation. On y trouvera notamment *httpd.conf* et *access.conf*.

## II.3. Installation du serveur Apache

### II.3.1. Installation à partir des packages

On doit disposer de deux packages [*apache\_1.3.14.XXXX* et *apache\_suexec\_1.3.14.XXXX*]. Le premier package est le serveur Apache, le second permet d'exécuter des scripts CGI en changeant le propriétaire du script CGI. La commande pour installer ces packages est la suivante :

```
rpm -ivh apache_1.3.14.XXXX  
rpm -ivh apache_suexec_1.3.14.XXXX
```

Le serveur est désormais installé.

### II.3.2. Installation à partir des sources

L'installation à partir des sources est un peu plus longue, elle se déroule en quatre étapes :

- Décompression de l'archive contenant les sources,
- Configuration,
- Compilation [Make],
- Installation [Make install].

#### - Décompression de l'archive

La décompression de l'archive s'effectue en tapant la commande suivante :

```
tar -zxf apache_1.3.14.tar.gz
```

Les fichiers décompressés se trouvent maintenant dans le répertoire */etc/httpd/apache*. Dans ce répertoire, il y a un fichier nommé *configure*, ce fichier va servir à la configuration du serveur apache.

#### - Configuration du serveur WEB

Simplement on tape */configure* et on laisse les options de configuration par défaut pour le serveur WEB.

### - La phase Compilation

Lorsque la phase de configuration s'achève sans erreur, la compilation peut débuter. Cette phase est très simple à mettre en oeuvre, il suffit de taper : ***make***

### - La phase Installation

Il suffit ici de taper : ***make install***  
L'installation du serveur web est terminée...

## II.3.3. Arborescence des répertoires

Le serveur Apache construit pour chaque machine correspondant à un "site web", quatre répertoires :

- \* La racine du serveur (ou site) où seront stockés les pages html : **htdocs** ou **html**.
- \* Le répertoire contenant les scripts : **cgi-bin**
- \* Le répertoire contenant les fichiers de configuration : **conf**
- \* Le répertoire contenant les fichiers "logs", historiques, accès et erreurs : **logs**

## II.4. Configuration d'Apache

La configuration d'Apache est définie dans trois fichiers au format texte dans le répertoire *conf*, que l'on peut éditer à tout moment. Il s'agit d'un certain nombre de variables à positionner, puis il suffit de relancer le serveur pour que les modifications soient prises en compte.

Ces variables ont une valeur par défaut, mais il s'agit surtout de savoir quelle variable on doit positionner pour répondre à un besoin bien précis. Un exemple de configuration (A ne pas utiliser tel quel) décrit l'objet de toutes les variables Ces 3 fichiers sont :

- **httpd.conf** : Spécification de tous les paramètres "techniques" du serveur . On donne le port d'utilisation, le nom DNS (à déclarer au préalable) du serveur, le chemin racine de l'emplacement du serveur le nom des fichiers de *LOG*...
- **srm.conf**: Configuration d'utilisation : format des données à partager , racine des documents HTML, possibilité d'utiliser les scripts CGI, les SSI, les méthodes connues (*GET POST*) , les comptes utilisateurs (*/~<nom>/*) ...
- **access.conf** : Liste des droits d'accès aux différentes parties de l'arborescence. On utilise des balises (même "style" que le HTML) pour définir une région bien précise de l'arborescence. Cependant, on peut redéfinir des permissions au niveau des répertoires à l'aide de fichier *.htaccess*.

Une fois Apache est installé , il faut le configurer comme ce qui suit :

- **ServerName** Définit le nom du serveur. Celui-ci sera accessible par ce nom. Dans notre cas on a choisit comme nom « *bibliothèque.tn* »
- **DocumentRoot** Cette directive définit le répertoire racine à partir duquel httpd va distribuer les fichiers. Dans notre cas c'est « */home/library/* »
- **DirectoryIndex** : Définit une liste de pages de démarrage pour le serveur exemple : *index.html*
- **<Directory> et </Directory>** : sont utilisés pour 'en capsuler' un groupe de directives applicables uniquement au répertoire indiqué ainsi qu'à ses sous-répertoires.
- Notre serveur s'appelle **bibliothèque.tn**, pour rendre accessible les fichiers html se trouvant sous **/home/library**, pour cela dans le fichier **/etc/httpd/conf/httpd.conf**, on va rajouter:

*Alias /icons/ "/etc/httpd/icons/"*

*Alias /index "/home/library/index.html"*

## II.5. Installation de PHP

L'installation de PHP se déroule comme suit :

*cd../Php-4\_x*

*/configure --enable-calendar --with-mysql --with-apache=../apache\_1.3.x*

Ensuite il faut compiler PHP :

*make*

*make install*

**- Configuration du serveur pour PHP**

Dans le fichier `/etc/httpd/conf/httpd.conf` et on valide la ligne de configuration :

*AddType application/x-httpd-php3.php3*

*AddType application/x-httpd-php.Php.Php4.phtml*

## Chapitre III

---

### Le SGBD MYSQL

---

#### III.1. Définition de MYSQL

MYSQL est un SGBDR(Système de Gestion des Bases de Données Relationnelles). Il est le plus utilisée chez les hébergeurs webs à cause de sa rapidité de sa fiabilité ainsi que le bis de ses nombreuses interfaces (PHP notamment). C'est un ensemble d'applications permettant non seulement mais aussi manipuler les données (ajout, suppression, modification et lecture), et de contrôler l'accès. C'est un logiciel serveur de bases de données multi-utilisateurs, multi-tâches qui offre de bonnes performances et supporte les bases de données relationnelles et le langage d'interrogation SQL. En terme de requêtes, son utilisation paraît proche de celle d'autres logiciels équivalant *Postgres ou Oracles*. Il est destiné à fonctionner dans un contexte de serveur, qui réalise des accès effectifs à la base et qui gère une file d'attente des diverses requêtes.

#### III.2. Organisation des données sous MYSQL

Les données sont structurées de la manière suivante :

- Un ensemble de données connexes regroupées en **enregistrement**.



Figure 5 : Données sous forme d'enregistrement

- L'ensemble d'enregistrements groupés dans une **table**.

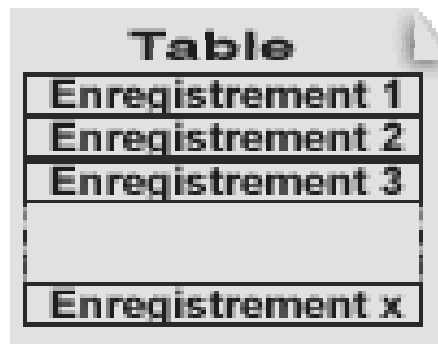


Figure 6 : Enregistrement sous forme de table

- L'ensemble de tables seront groupées dans **une base de données**.

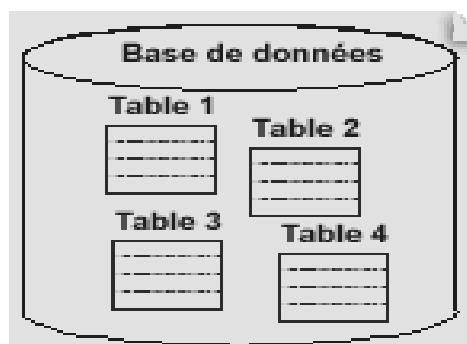


Figure 7 : Tables sous forme de base de données

- L'ensemble des bases de données composant le **serveur de données**.

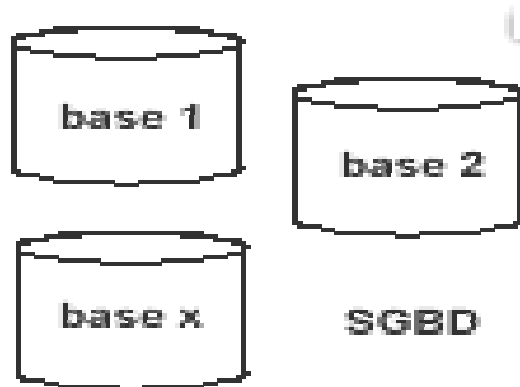


Figure 8 : Bases de données sous forme de SGBD

### III.3. Installation de MYSQL

Pour l'installation de MYSQL on distingue trois types de version d'installation :

#### III.3.1. Installation à partir de la version binaire (rpm)

Il faut télécharger une version binaire de MYSQL, c'est à dire un programme déjà compilé . La seule manipulation qu'on peut effectuer est la décompression du fichier par la commande **rpm** il est conseillé d'effectuer cette opération après la création d'un répertoire */usr/local/mysql*, dans lequel on peut placer le fichier.

#### III.3.2. Installation avec la version source rpm

Il existe des versions de MYSQL au format rpm qu'on peut télécharger sur le web. Plusieurs types de package sont disponibles, on prend le package nommé *MYSQL-3.23.36-1.src.rpm*. Ce package contient les sources de MYSQL, et il nous permettra d'installer la partie serveur et client de MYSQL .

Pour installer ce package, plusieurs alternatives sont offertes à vous : Utiliser un utilitaire graphique d'installation, plusieurs de ces utilitaires sont à la disposition (kpackage de KDE, rpmDrake distribué par Mandrake ,...) ou bien utiliser l'utilitaire rpm en ligne de commande.



Nous détaillerons ici l'installation en ligne de commande, l'installation en mode graphique se déroulant de la même manière.

On place le fichier `MYSQL-3.23.36-1.src.rpm` dans un répertoire quelconque puis on tape la commande suivante :

```
[root@localhost /root]# rpm -ivh --rebuild MYSQL-3.23.36-1.src.rpm
MYSQL #####
[root@localhost /root]#
```

Le fichier `MYSQL-3.23.36-1.rpm.gz` résultant de cette installation est alors placée dans le repertoire : `/usr/src/RPM/SOURCES` où il est archivé.

```
[root@localhost/root] # ll /usr/src/RPM/SOURCES total 4212
```

Puis on fait copier le fichier `mysql 3.23.36.rpm.gz` dans le répertoire HOME (`/root`).

### III.3.3. Installation à partir de la distribution source

Après la récupération d'une version source de MYSQL sous forme d'archive (extension **.tar.gz**), cet archive est une concaténation de fichier, extension (**.tar**) qu'on va compresser à l'aide du compresseur **gz**.

Cet archive sera placé dans un répertoire quelconque (répertoire d'accueil par exemple) : La commande **tar** sert à décompacté cet archive.

```
[root@localhost /root] # tar -xvzf mysql-3.23.36-1.tar.gz : Cette commande effectue deux choses : La décompression de l'archive et la reconstitution des fichiers concaténés ;
```

Un répertoire `mysql-3.23.36` à été crée dans le répertoire courant contenant toutes les sources de MYSQL.

```
[root@localhost /root] # cd mysql-3.23.36
```

La prochaine étape consiste à compiler l'ensemble des programmes à partir des fichiers sources, c'est à dire traduire les sources (ou programmes) en exécutables qui permettent d'utiliser MYSQL.

La compilation passe d'abord par la configuration du programme que l'on veut compiler à travers la commande **configure**. Cette commande peut prendre en paramètre le répertoire dans lequel sera installé le programme grâce à l'option **--prefix**

```
[Root/localhost /root] # ./configure --prefix=/usr/local/mysql
```

Durant cette procédure, le système est vérifié pour permettre une compilation normale (compilateur, librairies,...). La prochaine étape œuvre à la compilation effective du programme avec la commande :

```
[root@localhost /root]# make
```

puis la commande

```
[root@localhost /root] #make install
```

Cette commande va placer l'ensemble des programmes et fichiers nécessaires au bon fonctionnement de MySQL à l'endroit spécifié précédemment.

*La compilation de Mysql et son installation ont pour effet la création d'une arborescence dans le répertoire choisit lors de la procédure de la configuration(/usr/local/mysql).*

---

## Conclusion Générale

---

On ne peut clore ce mémoire sans mentionner le fait que le travail effectué, dans le cadre de ce projet de fin d'études nous a été bénéfique sur plusieurs niveaux.

En premier lieu, non seulement nous avons renforcé et enrichi nos connaissances dans le domaine de bases de données et Internet et ce par une étude bibliographique intense mais cela était également une occasion pour maîtriser et pratiquer les différents aspects de gestion de bases de données et d'implémentation d'un site web dynamique.

En second lieu, cette étude nous a permis de manipuler le système d'exploitation Linux, découvrir sa richesse et examiner de près cette plate-forme très utile pour les domaines informatiques et réseaux.

En troisième lieu, nous avons développé une interface base de données web ce qui nécessite l'étude des différentes techniques d'interfaçage, en particulier, PHP.

---

## Glossaire

---



**ASP** : Active Server Pages, ensemble de fichiers .asp et .html permettant l'accès aux bases de données.

**Apache** : Serveur Web configurable sous Linux.



**Balise** : Code employé dans les fichiers HTML, entre autres pour mettre en forme des documents destinés au WWW.



**CGI** : Common Gateway Interface, permet l'accès aux bases de données via des scripts appelés scripts CGI.

**Client** : Logiciel qui travaille pour le compte de l'utilisateur de manière à accéder à un serveur offrant un service quelque part sur le réseau.



**HTML** : (Hypertext Markup Language) : Langage de description des documents Web.

**HTTP:** (Hypertext Transfer Protocol) : Protocole de communication utilisé dans le web pour transférer une page HTML vers une poste client. C'est un protocole générique situé au niveau de la couche application.

**Hypertexte :** Un fichier texte contenant des liens extensibles vers d'autres documents situés soit sur le même ordinateur ou sur le réseau.



**Lien hypertexte :** C'est un moyen de passer d'un endroit à un autre par une simple clique.

**Linux :** Système d'exploitation multi-tâches et multi-utilisateurs.



**MYSQL :** Un système de gestion des bases de données relationnelles.



**PHP :** Professional HTML Preprocessing, langage de programmation qui met en évidence l'interface entre base de données et Web.



**Serveur :** Logiciel ou ordinateur permettant d'offrir des services ou des informations aux clients à la demande.

**SGBD :** Système de Gestion Base de Données.

**SQL :** Structured Query Language, langage permettant d'interroger et de mettre à jour une base de données relationnelle.

**Site :** Ensemble de pages web liées entre elles par des liens hypertexte ou des scripts applicatifs.

**Script :** Un programme exécuté côté serveur traitant les formulaires envoyés par le client.



**WEB :** Toile d'araignée, symbolise le réseau maillé de serveurs d'informations formant la toile d'araignée.

---

## Bibliographie

---

### Livres

- **Laurent Demarret, Fabrice Passero, Etienne Bataille , Dimitri Fontaine et Alexy Thomas**  
E-commerce sous Linux.  
Microapplication, août 2000.
- **Imad Saleh et Fabrice Papy**  
Les bases de données pour Internet et Intranet.  
Hermes, 1999.
- **Christian Marré et Guy Ledant**  
SQL, initiation, programmation et maîtrise.  
Armand Colin, 1989

### URLs visités :

- **www .Developpez . com**
- **www. mysql . com**
- **www. phpinfo . net**
- **www. php . net**
- **www. apache . com**
- **www . phpinfo . net**
- **www.nexen.net**

# **Implémentation d'une interface base de données web**

## **Réalisé par :**

Hamdi Mohamed Ali  
&  
Jabrane Khalil

## **Promotion :**

TS5 - Réseaux et Informatique

## **Résumé:**

Rendre les pages web dynamiques un domaine de recherche très vaste.

Dans ce projet on aentamé en premier lieu l'études des différents techniques qui peuvent jouer le rôle d'interface entre une source d'information,dans notre cas une base de données , et site web .

A la fin, on a mis en œuvre un type de site web dynamique, générant de l'informations cherchées par l'utilisateur, basé sur l'interaction entre le SGBD MYSQL et le serveur web Apache via des scripts PHP.

**Mots clés :** Linux, Apache, PHP, MYSQL, Web, Base de données.