

Haute École de la province de  
Liège,  
Catégorie technique

*2<sup>ieme</sup>* année de Bachelier en Techniques  
graphiques

Finalité Techniques infographiques

Introduction aux réseaux et aux bases de données

Support de cours provisoire

0.4



# Sommaire

<b>I</b>	<b>Introduction</b>	<b>1</b>
<b>1</b>	<b>Sites web statiques VS dynamiques</b>	<b>3</b>
1.1	Sites web statiques . . . . .	3
1.2	Sites web dynamiques . . . . .	3
1.3	Les différents types de sites dynamiques . . . . .	4
1.4	Mise en oeuvre d'un site dynamique . . . . .	5
<b>2</b>	<b>Publication de sites web</b>	<b>7</b>
2.1	Serveur web - Serveur HTTP - Sites web . . . . .	8
2.2	Serveurs dédiés - Serveurs mutualisés . . . . .	10
2.3	URL & DN . . . . .	12
2.4	Sous domaine . . . . .	14
2.5	Choisir un hébergeur . . . . .	16
2.6	Publication sur Internet . . . . .	18
<b>3</b>	<b>Droits d'accès des fichiers et dossiers sous UNIX</b>	<b>25</b>
3.1	User . . . . .	26
3.2	Group . . . . .	26
3.3	Attribution du user/group . . . . .	26
3.4	Droits . . . . .	27
3.5	Partager un dossier . . . . .	27
3.6	Cas concret du serveur Carl . . . . .	30
<b>II</b>	<b>SPIP</b>	<b>33</b>
<b>4</b>	<b>Découverte de SPIP</b>	<b>35</b>
4.1	Présentation de SPIP . . . . .	36

<b>5 Installation de SPIP</b>	<b>37</b>
5.1 Outils nécessaires . . . . .	38
5.2 Où placer les fichiers de spip . . . . .	38
5.3 Publication et installation de spip . . . . .	39
<b>6 Conception de squelettes</b>	<b>47</b>
6.1 Documentation SPIP . . . . .	48
6.2 Les balises propres au site . . . . .	48
6.3 Syntaxe des boucles . . . . .	49
6.4 Syntaxe complète des balises . . . . .	51
6.5 Les autres pages . . . . .	52
6.6 La balise #ENV . . . . .	54
6.7 Les filtres . . . . .	54
6.8 Secteur et branche . . . . .	57
6.9 Une boucle dans une boucle . . . . .	58
6.10 Compter le nombre d'élément d'une boucle . . . . .	59
6.11 GET & SET . . . . .	59
6.12 404, File not found . . . . .	59
6.13 Faire un menu correctement . . . . .	62
6.14 #EXPOSE . . . . .	65
6.15 Boucle récursive . . . . .	66
6.16 Les formulaires . . . . .	68
<b>III SGBD</b>	<b>71</b>
<b>7 Modèle relationnel &amp; SQL</b>	<b>74</b>
7.1 Introduction . . . . .	74
7.2 Le modèle relationnel . . . . .	75
<b>8 Comment placer une clé étrangère</b>	<b>94</b>
8.1 0,1 - 0,N . . . . .	95
8.2 1,1 - 0,N . . . . .	98
8.3 0,1 - 0,1 . . . . .	100
8.4 1,1 - 0,1 . . . . .	103
8.5 0,N - 0,N . . . . .	105
8.6 En résumé . . . . .	107
<b>9 SQL, manipulation de données</b>	<b>108</b>
9.1 SQL . . . . .	109
9.2 Projections . . . . .	113
9.3 Sélections . . . . .	115
9.4 Fonctions de calcul . . . . .	125
9.5 Opérateurs arithmétiques . . . . .	127
9.6 Dates et Heures . . . . .	128
9.7 Jointures . . . . .	130
9.8 Renommage - Alias . . . . .	135
9.9 ORDER BY . . . . .	136
9.10 GROUP BY . . . . .	137
9.11 Ajout de données . . . . .	139

9.12 Mise à jour de données . . . . .	141
9.13 Supprimer des données . . . . .	142
<b>Bibliographie</b>	<b>143</b>





# Introduction



# Chapitre

# 1

## Sommaire

- 1.1 Sites web statiques
- 1.2 Sites web dynamiques
- 1.3 Les différents types de sites dynamiques
- 1.4 Mise en oeuvre d'un site dynamique

# Sites web statiques VS dynamiques

## 1.1 Sites web statiques

On parle d'un site statique lorsque le contenu (les titres, les textes, ...) d'un site web est placé directement dans le code HTML. La moindre modification du contenu nécessite de modifier les fichiers HTML et donc d'en connaître l'organisation.

Un site web statique est donc un ensemble de fichiers HTML, CSS, d'objets multimédias (images, musiques, vidéos, ...) et éventuellement de code javascript pour animer le site web.

Un site statique convient très bien pour réaliser un site "carte de visite" qui ne contient que quelques pages et/ou le contenu ne change presque jamais.

Si le contenu évolue régulièrement (plusieurs fois par semaine ou par jour), ou que votre site contient un grand nombre de pages, la mise à jour du site va vite devenir difficile.

Si votre site web contient 50 pages, la moindre modification de la structure HTML (mise en page) demandera la modification des 50 pages. L'ajout d'une page demandera également l'ajout d'un lien vers cette nouvelle page dans plusieurs pages déjà existantes.

## 1.2 Sites web dynamiques

On parle de site web dynamique lorsque le contenu n'est pas stocké dans le code HTML mais dans des fichiers séparés ou mieux encore dans une base de données.

Ici les fichiers HTML constituent la structure du site et les contenus sont ajoutés à la volée par programmation. Ceci signifie que lorsque que l'on demande

au serveur la page d'accueil ou que l'on clique sur un lien, le serveur web remplit la structure HTML avec le bon contenu et l'envoi vers le navigateur.

Un site web dynamique est donc un programme qui peut être écrit presque dans n'importe quel langage (Php, Perl, Ruby, Python, Java, C#, C++, C, ...).

Pour créer un site dynamique, il est nécessaire d'écrire un site statique dans lequel les contenus seront remplacés par du code Php ou Spip par exemple.

Voici à quoi peut ressembler un site dynamique écrit en Php, il s'agit ici d'un exemple très basique.

```
<?php
...
// connexion à une base de données
...
// récupération de données en fonction d'un paramètre reçu de
// l'URL,
// le résultat est placé dans un tableau nommé data.
...
?>
<!DOCTYPE>
<html>
<head>
    <title> <?php echo $data['title']; ?> </title>
</head>
<body>
    <h1> <?php echo $data['titre']; ?> </h1>
    <p> <?php echo $data['chapo']; ?> </p>
    <a href="php echo $data['link']; ?&gt;"Lire la suite ...</a>
</body>
</html>
```

C'est un paramètre reçu par l'intermédiaire de l'URL qui déterminera le contenu de la page (nous en reparlerons plus tard).

L'avantage est que la présentation et le contenu sont séparés, une modification de la présentation du site est plus simple puisqu'il y a moins de fichiers à modifier : page d'accueil, page qui présente les articles, quelques pages spécifiques comme contact, a propos, ...

La mise à jour du contenu nécessite l'utilisation d'un outil supplémentaire ou la mise en place de formulaires.

On nomme « frontend » (partie publique) les pages qui permettent de présenter les données et « backend » (partie privée) les pages qui permettent de mettre à jour le contenu.

Ici, il faut donc programmer (beaucoup) pour avoir un site opérationnel. Heureusement, il existe un grand nombre de moteur de site web dynamique prêt à l'emploi.

### 1.3 Les différents types de sites dynamiques

Les sites web dynamiques sont en fait des logiciels de gestion de contenu, que l'on nomme CMS en anglais pour « Content Management System ».

Certains CMS sont très spécialisés et adaptés à une tâche bien précise.

**Les blogs** sont des sites que vous rencontrez tous les jours lorsque vous surfez sur le net. Il s'agit d'une simple liste d'articles affichés de manière anti-chronologique. Les articles peuvent être classés par catégorie. Les deux moteurs de blog les plus connus sont : Wordpress et dotclear.

**Les forums** ne sont plus à présenter. Le moteur le plus connu est phpBB.

**Les e-commerces** sont des sites spécialisés pour la création de site de vente en ligne. Ici, il est possible de gérer un stock, le prix des articles. On retrouve ici Prestashop, Magento, os-commerce, le plugin Virtumart de Joomla et bien d'autres.

**Les elearnings** permettent de concevoir facilement des questionnaires et d'avoir un système de correction automatique. Dans le cas d'une mauvaise réponse, de renvoyer vers une page de théorie.

**Les galeries d'images** ont également des moteurs dédiés, les plus connus sont : Coppermine et PhpWebGallery.

**Les CMS généralistes** permettent de gérer de manière très souple le contenu d'un site. Ceci a un coût, la prise en main d'un tel CMS est généralement moins simple. On retrouve ici des sites comme Joomla, Drupal ou SPIP.

On peut encore ajouter à cette liste les wikis, les groupwares, les gestionnaires de calendriers, les gestionnaires de bugs. Le site [www.opensourcecms.com](http://www.opensourcecms.com) présente plus de 300 CMS que vous pouvez tester en ligne sans devoir les installer vous-même.

Le succès du moteur de blog wordpress a fait qu'un grand nombre de plug-ins a été développé pour étendre ses capacités initiales. Il est donc parfois difficile de placer un logiciel dans une catégorie plutôt qu'une autre.

## 1.4 Mise en oeuvre d'un site dynamique

Pour créer un site statique, un éditeur de texte et un navigateur suffisent. Dans le cas des sites dynamiques, comme il s'agit de programmes, vous devez absolument disposer d'un serveur web capable d'exécuter le code du moteur.

Généralement, les moteurs web sont écrits en Php et utilisent MySql (maintenant on devrait dire MariaDB) comme base de données, le serveur web étant généralement Apache. On parle d'architecture LAMP sous Linux, WAMP sous Windows et MAMP sous MacOS.

Le développement s'effectue en local sur notre machine et l'on publie le site seulement lorsqu'il est terminé. Il faut donc installer sur votre machine une architecture xAMP.

Soit vous installez et configurez ces outils vous même, soit vous utilisez un toolkit prêt à l'emploi.

Sous Windows, il existe :

- WAMP Server  <http://www.wampserver.com>
- easyPhp  <http://www.easyp.php.org>
- XAMPP  <https://www.apachefriends.org>

Sous MacOS, il existe :

- MAMP  <http://www.mamp.info>
- XAMPP  <https://www.apachefriends.org>

Sous Linux, cela dépend de votre distribution.

Dans le cas de l'école :

- Soit vous utilisez votre portable et installez un toolkit.
- Soit vous utilisez le serveur de développement de l'école et vos fichiers sont en permanence sur le serveur (ici ce n'est pas un serveur de production ce n'est pas gênant).
- Soit vous utilisez WAMP sur une machine de laboratoire, mais alors vous devrez sauvegarder à la fin de chaque cours vos fichiers HTML et votre base de données, mais également les restaurer au début de cours.

## **Chapitre**

# **2**

### **Sommaire**

- 2.1 Serveur web - Serveur HTTP - Sites web
- 2.2 Serveurs dédiés - Serveurs mutualisés
- 2.3 URL & DN
- 2.4 Sous domaine
- 2.5 Choisir un hébergeur
- 2.6 Publication sur Internet

## **Publication de sites web**

Ce chapitre reprend une série de notions des réseaux informatiques liées à la publication de sites web.

## Sommaire

- ① Site web statique vs Site web dynamique
- ② Serveur web - Serveur HTTP - Sites web
- ③ Serveurs dédiés - Serveurs mutualisés
- ④ URL & DN
- ⑤ Sous domaine
- ⑥ Choisir un hébergeur
- ⑦ Publication sur Internet

## Serveur Web vs Serveur HTTP

Un logiciel serveur :

- est un programme qui offre un service (exp : échange d' email) ;
- c'est un programme qui attend qu'on lui demande quelque chose ;
- en général, il ne possède pas d'interface graphique et est démarré en tâche de fond ;
- sous Windows, on parle de service (service FTP, service HTTP, service SMTP, ...) ;
- sous UNIX, on parle de "daemon" (FTPd, HTTPd, CUPSd, ...).

## Serveur Web vs Serveur HTTP

Un serveur HTTP :

- est un programme qui répond à des requêtes suivant le protocole HTTP ;
- HTTP est un protocole de transfert de fichiers HyperText ;
- permet donc de partager le contenu d'un dossier déterminé.

Un serveur Web :

- c'est un ordinateur qui propose un service HTTP ;
- généralement, il est accompagné par MySql et par PHP.

## Site web

Un site web, c'est :

- soit le dossier partagé par le serveur HTTP,  
c :\www → http ://www.petzi.be
- soit un sous-dossier du dossier partagé,  
c :\www\user04 → http ://www.free.fr/user04  
c :\www\user640 → http ://www.free.fr/user640

Serveur web - Serveur HTTP - Sites web

## Boîte à outils

**LAMP** Linux Apache MySql Php  
→ dépend de votre distribution.

**MAMP** Mac Apache MySql Php  
→ Mamp 2.0.3

**WAMP** Windows Apache MySql Php  
→ wampserver 2.1e

F. Gerard (HEPL)

Intro R&B

2014-2015

13 / 40

Serveurs dédiés - Serveurs mutualisés

## Sommaire

- ① Site web statique vs Site web dynamique
- ② Serveur web - Serveur HTTP - Sites web
- ③ Serveurs dédiés - Serveurs mutualisés
- ④ URL & DN
- ⑤ Sous domaine
- ⑥ Choisir un hébergeur
- ⑦ Publication sur Internet

F. Gerard (HEPL)

Intro R&B

2014-2015

14 / 40

## Mutualisé vs Dédié

### Hébergement mutualisé :

lorsque des sites web se trouvent sur un même serveur.

#### Avantage :

- moins onéreux

#### Inconvénient :

- les ressources sont partagées entre plusieurs sites.  
Un site trop gourmand en ressource pourrait ralentir votre site.

## Mutualisé vs Dédié

### Hébergement dédié :

lorsque un serveur est utilisé exclusivement pour un site web.

#### Avantage :

- toutes les ressources du serveur sont destinées à l'exploitation du site web.

#### Inconvénient :

- plus onéreux qu'un service mutualisé.

## Sommaire

① Site web statique vs Site web dynamique

② Serveur web - Serveur HTTP - Sites web

③ Serveurs dédiés - Serveurs mutualisés

④ URL & DN

⑤ Sous domaine

⑥ Choisir un hébergeur

⑦ Publication sur Internet

## URL & DN

**URL** Uniform Resource Locator,  
indique comment retrouver une ressource sur un réseau.  
Exemples :

- <http://www.petzi.be/img/header.jpg>
- <http://www.petzi.be/forum>

**DN** Domain Name - nom de domaine,  
permet d'utiliser un nom à la place d'une IP  
Exemple : petzi.be

## Idées reçues ? !

- 1 nom de domaine remplace 1 et 1 seule adresse IP ? **VRAI**
- 1 adresse IP désigne 1 et 1 seule machine sur Internet ? **VRAI**

==>

- Des sites web dont le DN est  $\neq$  sont-ils hébergés sur des serveurs différents ? (par exemple: yoda.org ; vader.net ; palpatine.com ; ...)  
==>**PAS FORCLEMENT**

- On peut faire “pointer” deux DN  $\neq$  vers une même IP.
- Un ordinateur peut posséder plusieurs adresses IP.

## Sites web VS Serveurs web

On a donc PAS 1 site web = à 1 serveur web

- `http://www.free.fr/user04`
  - `http://www.free.fr/user640`
- 
- `http://www.yoda.org`
  - `http://www.vader.net`

Sous domaine

## Sommaire

① Site web statique vs Site web dynamique

② Serveur web - Serveur HTTP - Sites web

③ Serveurs dédiés - Serveurs mutualisés

④ URL & DN

⑤ Sous domaine

⑥ Choisir un hébergeur

⑦ Publication sur Internet

F. Gerard (HEPL)

Intro R&B

2014-2015

21 / 40

Sous domaine

## URL & DN

[http ://www.petzi.be/blog](http://www.petzi.be/blog)

[http ://](http://) désigne le protocole, le langage utilisé par le client et le serveur pour communiquer.

[petzi.be](http://petzi.be) c'est le nom de domaine.

[www](http://www.petzi.be) est là par convention.

[blog](http://www.petzi.be/blog) désigne le dossier blog du serveur petzi.be.

F. Gerard (HEPL)

Intro R&B

2014-2015

22 / 40

Sous domaine

## URL & DN

http ://blog.petzi.be

blog est ce que l'on nomme un sous domaine.

Un sous domaine peut pointer vers un dossier du serveur petzi.be,

ou bien peut pointer vers un autre serveur.

Sous domaine

## URL & DN

ftp ://ftp.petzi.be

ftp :// File Transfert Protocoll,  
Protocole de Transfert de Fichiers.

ftp. devant le petzi.be est aussi une convention.

Choisir un hébergeur

## Sommaire

① Site web statique vs Site web dynamique

② Serveur web - Serveur HTTP - Sites web

③ Serveurs dédiés - Serveurs mutualisés

④ URL & DN

⑤ Sous domaine

⑥ Choisir un hébergeur

⑦ Publication sur Internet

F. Gerard (HEPL)

Intro R&B

2014-2015

25 / 40

Choisir un hébergeur

## Comparons

- ovh.com
- one.com
- planethoster.com

F. Gerard (HEPL)

Intro R&B

2014-2015

26 / 40

Choisir un hébergeur

## Identifiants relatifs à un hébergement

Pour le domaine exemple.net, on pourrait avoir ceci :

control panel login : exemple.net  
password : \*\*\*

compte FTP hôte : ftp.exemple.net  
login : exemple.net  
password : \*\*\*

MySql hôte : exemple.net.mysql  
base de données : exemple.net  
login : exemple.net  
password : \*\*\*

Choisir un hébergeur

## Identifiants relativ à un hébergement

PhpMyAdmin https://dbadmin.one.com  
login : exemple.be  
password : \*\*\*

Émail serveur IMAP : imap.exemple.net  
serveur POP3 : pop.exemple.net  
serveur SMTP : send.one.com  
webmail : https://www.one.com

## Sommaire

- ① Site web statique vs Site web dynamique
- ② Serveur web - Serveur HTTP - Sites web
- ③ Serveurs dédiés - Serveurs mutualisés
- ④ URL & DN
- ⑤ Sous domaine
- ⑥ Choisir un hébergeur
- ⑦ Publication sur Internet

## Apache

Un serveur apache reprend, entre autre, les dossiers suivants :

`conf` contient le fichier de configuration “`httpd.conf`”.

`conf/extra` contient des fichiers de configuration supplémentaire

`logs` fichiers de log, contiennent des informations sur ce que fait le serveur. C'est très utile si le serveur plante. On peut y trouver un message d'erreur qui permettra de comprendre ce qu'il se passe.

`htdocs` est le dossier principale contenant les documents partagés par le serveur.

## Publication sur Internet

### Apache

- Dans la représentation de chacun, un ordinateur héberge un seul site web (on a vu plus haut que ce n'est pas toujours le cas).
- Au vue des performances des machines actuelles, pour de petits sites web, cela sous exploiterait les performances des machines serveurs.
- On a donc la possibilité de configurer Apache pour réaliser un hébergement mutualisé multi-domaine.
- Cette technique se nomme virtual host.
- En gros, pour chaque domaine hébergé par apache, on définira un dossier htdocs.

## Publication sur Internet

### FTP

Il faut pouvoir copier les fichiers HTML et autres dans le dossier du serveur HTTP qui vous est destiné.

- Soit vous avez un accès physique à la machine et vous pouvez copier directement les fichiers depuis un stick USB.
- Soit vous devez copier les fichiers par Internet via FTP.

l'hébergeur fait en sorte que le dossier atteint par FTP (ftp.exemple.net) coïncide avec celui atteint par votre navigateur (www.exemple.net).

## Transfert FTP

FTP : File Transfert Protocol

- Permet l'échange de fichiers entre :
  - un client et un serveur ;
  - deux serveurs.
- Utilise deux canaux de communication :
  - Le 1ier établi par le client vers le serveur, pour l'échange des commandes (copier, supprimer, change de dossier, ...);
  - Le 2ème est utilisé pour l'échange des données.

## Transfert FTP

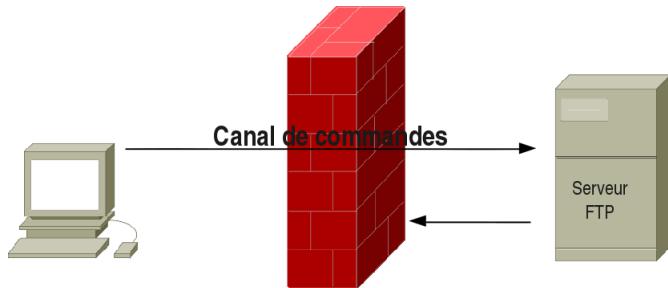
Il existe 2 modes pour ouvrir le canal de données.

- Mode actif, c'est le serveur qui ouvre le canal vers le client.
- Mode passif, c'est le client qui ouvre le canal vers le serveur.

Publication sur Internet

## Transfert FTP

mode actif

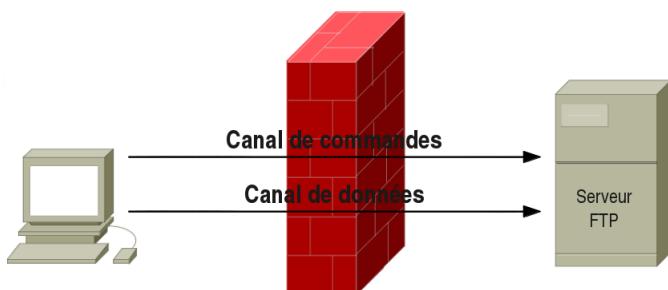


Lorsque vous êtes derrière un firewall, le serveur ne sait pas ouvrir le canal de données (sauf si vous configurez votre firewall!).

Publication sur Internet

## Transfert FTP

mode passif



Ici, c'est le client qui ouvre les deux canaux de communication.





Le gestionnaire de site permet de conserver plusieurs configurations.



Chaque site peut être configuré individuellement.



## Chapitre

# 3

### Sommaire

- 3.1 User
- 3.2 Group
- 3.3 Attribution du user/group
- 3.4 Droits
- 3.5 Partager un dossier
- 3.6 Cas concret du serveur Carl

## Droits d'accès des fichiers et dossiers sous UNIX

La plupart des serveurs web sont des machines UNIX, il est donc important de comprendre et maîtriser la gestion des droits d'accès sur ce type de système.

Il y a trois notions qui entrent en jeu pour gérer les droits d'accès à un fichier ou à un dossier.

### 3.1 User

La première est la notion de user (utilisateur), chaque personne travaillant sur une machine UNIX est identifié par son user (login). Si vous vous connectez avec le login "fred", vous serez donc l'utilisateur "fred" pour le système. Lorsque vous lancez un programme celui-ci s'exécute sous votre identité.

Lorsqu'un programme est démarré automatiquement au démarrage de la machine, il lancé avec l'identité de l'administrateur (nommé root sous UNIX). Lorsque le serveur web exécute du code PHP par exemple, le code est alors exécuté sous l'identité de l'administrateur. Cela signifie que vous pouvez faire tout et n'importe quoi sur ce serveur (vous avez les droits admin) !

Pour cette raison, les programmes serveurs doivent être configurés pour s'exécuter avec un utilisateur fictif. Par fictif, on entend : si http est le nom d'utilisateur attribué au serveur web, il n'est pas possible de ce logger au serveur (ouvrir une session) avec l'utilisateur http.

### 3.2 Group

La seconde notion est la notion de groupe (group en anglais). Chaque utilisateur d'une machine UNIX est repris dans un ou plusieurs groupes d'utilisateurs.

Si l'on donne un droit à un groupe, tous les utilisateurs faisant partie de ce groupe possèdent ce droit.

En général, lorsque l'administrateur de la machine Linux crée un utilisateur, il crée également un groupe portant le même nom. Il s'agit de votre groupe par défaut.

### 3.3 Attribution du user/group

A chaque fichier (ou dossier) est assigné un utilisateur (le propriétaire) et un groupe.<sup>1</sup> Lorsque vous créez un fichier (ou un dossier), vous en êtes automatiquement le propriétaire. Le groupe assigné au fichier (ou dossier) sera votre groupe par défaut, qui porte généralement le même nom que votre nom d'utilisateur.

Voici un exemple de fichiers et dossiers :

fileName	user	group
fichier1.txt	fred	fred
dossier1	fred	fred
fichier2.txt	fred	users
dossier2	fred	users
fichier3.txt	tom	tom
dossier3	tom	users

---

1. Plus exactement, le user/group attribué à un fichier (ou dossier) est le userid et le groupid de l'application/user qui le créé.

Si vous êtes le propriétaire d'un fichier (ou dossier), vous pouvez changer le groupe de ce fichier (ou dossier). Si vous êtes le super administrateur (root), vous pouvez changer le propriétaire et le groupe de ce que vous voulez. Sur un serveur web, il n'est généralement pas nécessaire de modifier l'utilisateur ou le groupe de vos fichiers ou dossiers.

### 3.4 Droits

Il y a trois types de droits :

- le droit de lecture, noté "r" :
  - dans le cas d'un fichier, il permet de lire le fichier ;
  - dans le cas d'un dossier, il permet de lister le dossier ;
- le droit d'écriture, noté "w" :
  - dans le cas d'un fichier, il permet de modifier/supprimer le fichier ;
  - dans le cas d'un dossier, il permet d'écrire dans le dossier (créer des sous-dossiers, des fichiers, ...),  
si vous avez le droit de supprimer tous les fichiers et sous-dossiers contenus, vous pouvez supprimer ce dossier.
- le droit d'exécution, noté "x" :
  - dans le cas d'un fichier, il permet d'exécuter le fichier (qui doit évidemment être un programme, sinon cela n'a pas de sens) ;
  - dans le cas d'un dossier, il permet de rentrer dans le dossier.

Voici un exemple de fichiers et dossiers :

fileName	user	group	rights
fichier1.txt	fred	fred	rwx rwx r-
dossier1	fred	fred	rwx ---
fichier2.txt	fred	users	rwx --- r-
dossier2	fred	users	r-x r-x r-x
fichier3.txt	tom	tom	r- r- r-
dossier3	tom	users	rwx --- r-x

Chaque fichier (ou dossier) possède des droits rwx pour le propriétaire, rwx pour le groupe attribué au fichier (ou dossier) et rwx pour les autres (ceux qui ne font pas partie du groupe). Les droits sont toujours donnés dans cet ordre.

Dans l'exemple ci avant, pour le fichier « le fichier1.txt », le propriétaire peut lire et écrire, les utilisateurs du groupe peuvent également lire et écrire dans le fichier, les autres utilisateurs du système ne peuvent que lire le fichier. Le “-” montre l'absence d'un droit.

### 3.5 Partager un dossier

Prenons comme exemple le dossier /home qui contient les dossiers des utilisateurs :

fileName	user	group	rights
/home/fred	fred	fred	rwx — —
/home/tom	tom	tom	rwx — —

Ici seul l'utilisateur fred peut accéder au dossier /home/fred, de même que tom ne peut accéder qu'à son dossier.

Maintenant, nous allons chercher à partager un sous-dossier de fred avec tom.

## 1ier méthode

Une première manière de procéder est de donner un droit de lecture et d'exécution au groupe fred et de mettre tom dans le groupe fred. Les droits deviennent ceci :

fileName	user	group	rights
/home/fred	fred	fred	rwx r-x —
/home/tom	tom	tom	rwx — —

Le problème est que l'administrateur n'ajoutera pas tom dans votre groupe (il n'a pas que cela à faire). De plus, si votre groupe contient 40 utilisateurs, et que seul tom doit pouvoir accéder à ce dossier, alors les 39 autres utilisateurs peuvent également y accéder.

## 2ème méthode

Une deuxième manière de faire est de changer le groupe du fichier en tom, les droits deviennent ceci :

fileName	user	group	rights
/home/fred	fred	tom	rwx r-x —
/home/tom	tom	tom	rwx — —

Cette méthode a pour inconvénient que tom (et tous les membres du groupe tom) peut voir tout le contenu de votre dossier personnel.

On va donc lui permettre de rentrer dans votre dossier mais pas de voir son contenu. les droits deviennent ceci :

fileName	user	group	rights
/home/fred	fred	tom	rwx -x —
/home/tom	tom	tom	rwx — —

Maintenant que les membres du groupe tom peuvent rentrer dans le dossier de fred, il faut regarder les droits des fichiers et dossiers à l'intérieur de /home/fred.

fileName	user	group	rights
/home/fred/top_secret/	fred	fred	rwx rwx rwx
/home/fred/share_tom/	fred	fred	rwx rwx rwx
/home/fred/carte_visa.txt	fred	fred	rw- rw- rw-

Ici, tout est en accès libre. Les membres du groupe tom ne voient pas vos fichiers, mais s'ils tapent des noms de fichiers (ou de dossiers) au hasard, il est possible qu'ils ouvrent ce qu'ils ne doivent pas.

On doit donc régler les droits dans /home/fred comme ceci :

fileName	user	group	rights
/home/fred/top_secret/	fred	fred	rwx rwx —
/home/fred/share_tom/	fred	fred	rwx rwx —
/home/fred/carte_visa.txt	fred	fred	rw- rw- —

Le but est que tom puisse aller lire dans le dossier share\_tom, on doit donc adapter le groupe et les droits de ce dossier :

fileName	user	group	rights
/home/fred/top_secret/	fred	fred	rwx rwx —
/home/fred/share_tom/	fred	tom	rwx r-x —
/home/fred/carte_visa.txt	fred	fred	rw- rw- —

Attention, tous les membres du groupe tom peuvent accéder au sous-dossier share\_tom.

### 3ème méthode

Il se trouve maintenant que je veux partager un second dossier avec un troisième utilisateur, hugo.

Je ne sais pas utiliser la même méthode puisque un dossier (ou fichier) ne possède qu'un groupe. Il est donc impossible que le dossier /home/fred soit assigné aux groupes tom et hugo.

Je vais donc placer les droits comme ceci sur le dossier /home/fred

fileName	user	group	rights
/home/fred	fred	fred	rwx — -x
/home/tom	tom	tom	rwx — —

A la place de :

fileName	user	group	rights
/home/fred	fred	tom	rwx -x —
/home/tom	tom	tom	rwx — —

Je permet ainsi à tous les utilisateurs de rentrer dans mon dossier personnel, mais ils ne peuvent pas le lister.

Il faut donc veiller à ce que le contenu de mon dossier personnel ne soit pas accessible pour les "autres". Il faut attribuer le groupe tom pour le dossier share\_tom et le groupe hugo pour le dossier share\_hugo tous deux avec les droits r-x.

fileName	user	group	rights
/home/fred/top_secret/	fred	fred	rwx r-x --
/home/fred/share_tom/	fred	tom	rwx r-x --
/home/fred/share_hugo/	fred	hugo	rwx r-x --
/home/fred/public	fred	fred	rwx r-x r-x
/home/fred/carte_visa.txt	fred	fred	rw- r- --

Il est évident que tom et hugo doivent connaître le nom du dossier que je leur destine.

Le dossier public, lui, serait accessible à tous.

### 3.6 Cas concret du serveur Carl

Carl (10.59.26.130) est le serveur web que nous utilisons dans le cadre de notre laboratoire. Chaque étudiant dispose d'un user nommé userxxx, où xxx est un nombre compris entre 1 et 200.

Votre dossier personnel sur Carl est donc /home/userxxx. Les droits d'accès à ce dossier ont déjà été réglé correctement. L'idée est de créer un dossier nommé public\_html et de partager ce dossier avec l'utilisateur http.

La raison est que le serveur web installé sur Carl utilise le user nommé http pour accéder aux fichiers et dossiers. Par analogie avec la 3ème méthode présentée ci-dessus, on va régler les droits en :

fileName	user	group	rights
/home/userxxx	userxxx	userxxx	rwx -- -x
/home/userxxx/public_html/	userxxx	http	rwx r-x --
/home/userxxx/carte_visa.txt	userxxx	userxxx	rw- -- --

Par facilité on peut régler les droits du dossier public\_html en : « userxxx userxxx rwx -- r-x ». Mais attention, tous les utilisateurs connectés au serveur peuvent copier vos fichiers.

Pas grave me direz-vous, on peut déjà les copier depuis le navigateur. Oui, mais pas les fichiers Php. Lorsque l'on demande un fichier php à un serveur web, il est interprété et l'on ne reçoit qu'un résultat en HTML.

Hors il y a presque toujours un fichier Php qui contient au minimum le login et password de la base de données. Autant sécuriser au mieux vos fichiers.

Un esprit tordu peut penser à écrire un programme Php qui parcourerait les dossiers /home/xxx/public\_html à la recherche de fichiers php contenant le login/password des bases.

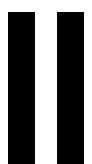
Un serveur bien configuré à prévu cela, notamment en confinant chaque script dans son dossier home. Autrement dit, il n'est pas possible de remonter plus haut que son dossier /home/userxxx. Le contenu du dossier public\_html peut avoir les droits suivants :

fileName	user	group	rights
public_html/spip	userxxx	userxxx	rwx — r-x
public_html/designweb	userxxx	userxxx	rwx — r-x

Seul le propriétaire et le groupe http peuvent accéder au dossier public\_html, donc les droits internes peuvent être moins restrictifs.

Ces changements de droits sont à effectuer par FTP ou par ssh.





**SPIP**



## Chapitre

# 4

## Découverte de SPIP

Ce chapitre reprend une série de liens vers la documentation officielle de SPIP. La documentation de SPIP est assez volumineuse, le but est donc de vous la faire découvrir facilement en vous proposant un nombre limité d'articles dans un ordre choisi.

## 4.1 Présentation de SPIP

Spir est un système de gestion de contenu (CMS : Content Management System), il permet de mettre en ligne une site web du type magazine.

SPIP est abondement documenté sur le site officiel <http://www.spip.net/fr>, mais également sur <http://spippourlesnuls.fr>.

### 4.1.1 C'est quoi SPIP ?

C'est quoi SPIP ?

• [http://www.spip.net/fr\\_article464.html](http://www.spip.net/fr_article464.html)

Caractéristiques complètes

• [http://www.spip.net/fr\\_article890.html](http://www.spip.net/fr_article890.html)

C'est quoi cette histoire d'espace privé ?

• [http://www.spip.net/fr\\_article1471.html](http://www.spip.net/fr_article1471.html)

Quels sont les éléments gérés par SPIP ?

• [http://www.spip.net/fr\\_article497.html](http://www.spip.net/fr_article497.html)

Le statut des auteurs dans SPIP

• [http://www.spip.net/fr\\_article3517.html](http://www.spip.net/fr_article3517.html)

### 4.1.2 Fonctionnement de SPIP

Guide du rédacteur

• [http://www.spip.net/fr\\_rubrique483.html](http://www.spip.net/fr_rubrique483.html)

Principe général

• [http://www.spip.net/fr\\_article3437.html](http://www.spip.net/fr_article3437.html)

Où placer les fichiers de squelettes ?

• [http://www.spip.net/fr\\_article3347.html](http://www.spip.net/fr_article3347.html)

Structure générale de SPIP

• [http://www.spip.net/fr\\_article3634.html](http://www.spip.net/fr_article3634.html)

## Sommaire

- 5.1 Outils nécessaires
- 5.2 Où placer les fichiers de spip
- 5.3 Publication et installation de spip

## Chapitre

# 5

## Installation de SPIP

SPIP est un logiciel web écrit en Php. Il est donc indispensable de disposer d'un serveur web capable d'exécuter du code Php.

## 5.1 Outils nécessaires

Pour utiliser SPIP, il faut disposer d'un espace web (hébergement) ayant les caractéristiques minimums suivantes :

- Serveur web Apache 1.2 ou compatible ;
- Php 5.1.0
- MySQL 4, MySQL 5 ou SQLite3

Soit, vous utilisez un hébergement distant (sur un serveur) et vous devez utiliser un client FTP (comme FileZilla) pour copier vos fichiers sur le serveur.

Soit, vous travaillez en local (sur votre ordinateur) en installant un toolkit (voir page 5).

La seule différence, entre hébergement local et distant, est :

- en local, il n'y a pas besoin de client FTP. Vous copiez simplement les fichiers dans un dossier particulier de votre machine.
- en distant, vous devez copier vos fichiers sur le serveur après chaque modification.

## 5.2 Où placer les fichiers de spip

### 5.2.1 En local

Sous Windows, en fonction de votre installation, vous devez placer vos fichiers soit

- dans C:\wamp\www,
- dans C:\Programmes\wamp\www,
- dans c:\Program Files (x86)\wamp\www.

Sous MacOS X, vous devez placer vos fichiers dans /Applications/MAMP/htdocs.

Sous Linux, cela dépend de votre installation, mais généralement, vous pouvez placer vos fichiers comme pour un serveur distant.

### 5.2.2 En distant

Sur un hébergement payant, vous disposerez d'un dossier nommé, soit www, soit public\_html. Les droits d'accès à ce dossier seront configurés correctement.

Sur le serveur Carl, vous devrez créer un dossier nommé public\_html à la racine de votre dossier personnel. Vous devrez lui appliquer les droits 705. Votre dossier personnel devrait déjà être configuré correctement (en 701).



Si vous utilisez l'installateur automatique de SPIP, c'est le serveur qui copie les fichiers. Donc ces fichiers ne vous appartiennent pas et vous ne pouvez pas les supprimer par FTP ou ssh. Copiez mon script *deltree.php* dans le dossier à supprimer et lappelez le via le navigateur. Si le dossier à éliminer se nomme spip-auto, exécutez le script par <http://10.59.26.130/~userxx/spip-auto/deltree.php>

## 5.3 Publication et installation de spip

Vous trouverez un guide à cette url [http://www.spip.net/fr\\_article5521.html](http://www.spip.net/fr_article5521.html) et également à celle-ci <http://spippourlesnuls.fr/?-INSTALLER-SPIP,30->.

Ce qui suit est une copie des slides avec les informations concernant le serveur de l'école.

### ÉTAPES DE PUBLICATION

- Télécharger «spip.zip»
- Décompresser l'archive en local
- Par FTP, transférer l'archive décompréssée sur le serveur Carl
- Par Web, configurer spip

### PARAMÈTRES FTP

- Configurer le mode «actif»
- Adresse I.P du serveur Carl : 10.59.26.130
- Login : userXX
- Password : \*\*\*\*

## FTP - FILEZILLA

- Une fois connecté sur le serveur vous êtes dans le dossier «/home/userXX»
  - Le serveur HTTP (Apache) est configuré pour accéder à «/home/userXX/public\_html»
  - Vous devez donc créer ce dossier
  - Vous pouvez ensuite uploader le dossier de spip dans «public\_html», mais ...

## COMMENT COPIER SON SITE

- Soit vous utilisez 1 seul moteur de site web (Spip, Wordpress, PhpBB, ...)

=> Copier le contenu de l'archive directement dans «public\_html»

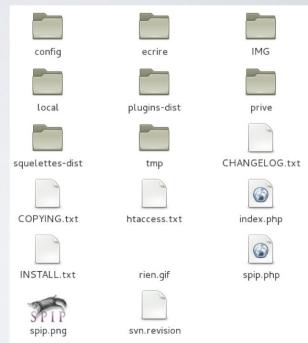
Attention l'archive se déballe dans un dossier «spip», c'est le contenu du dossier qu'il faut transférer

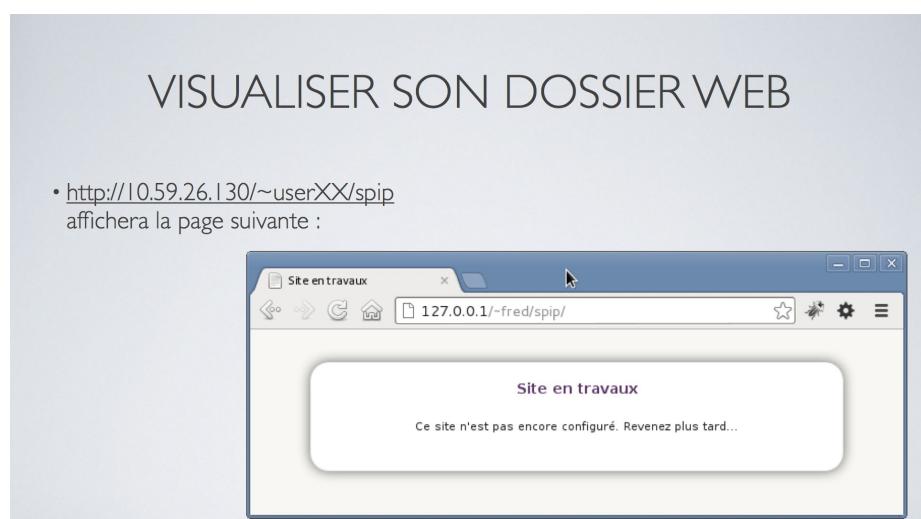
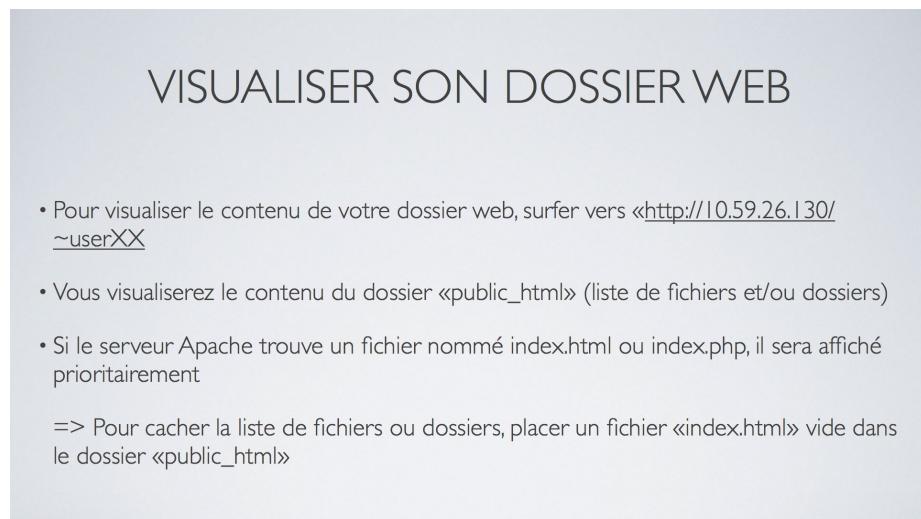
- Soit vous souhaitez installer plusieurs applications différentes (par exemple Spip + PhpBB)

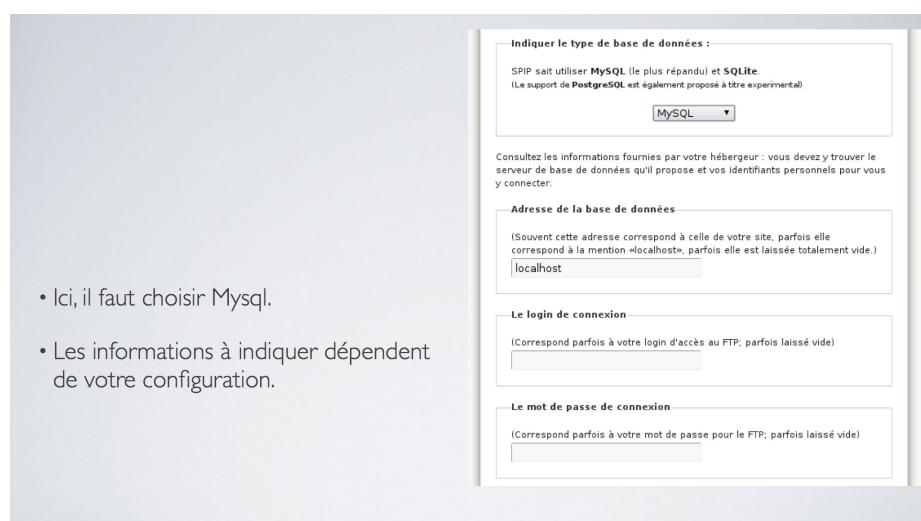
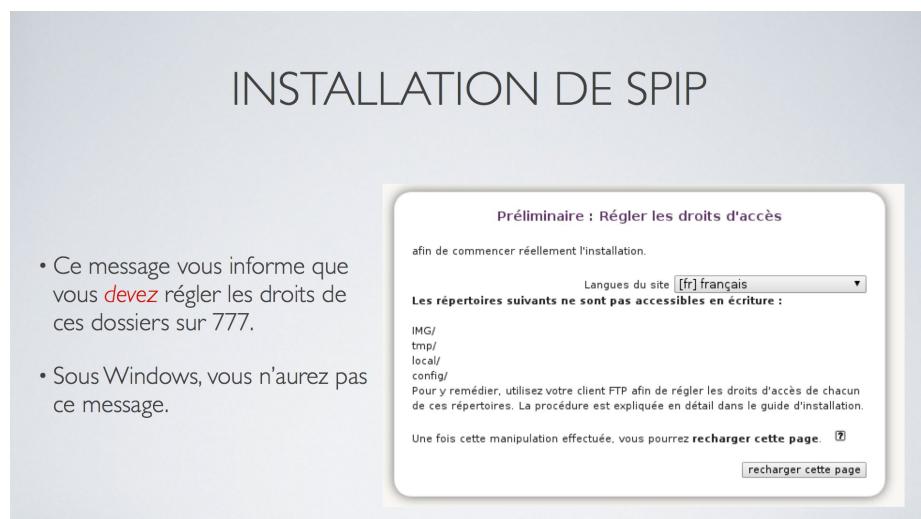
=> Copier le contenu de l'archive dans un sous-dossier de «public\_html»

# COMMENT COPIER SON SITE ?

- Dans notre cas, nous souhaitons pouvoir installer plusieurs copies de Spip ou d'autre logiciel web
  - Pour ce qui suit, j'installerai Spip dans le sous-dossier «spip»
  - Le contenu de .../public\_html/spip doit correspondre à l'image de droite







## LOCALHOST ?

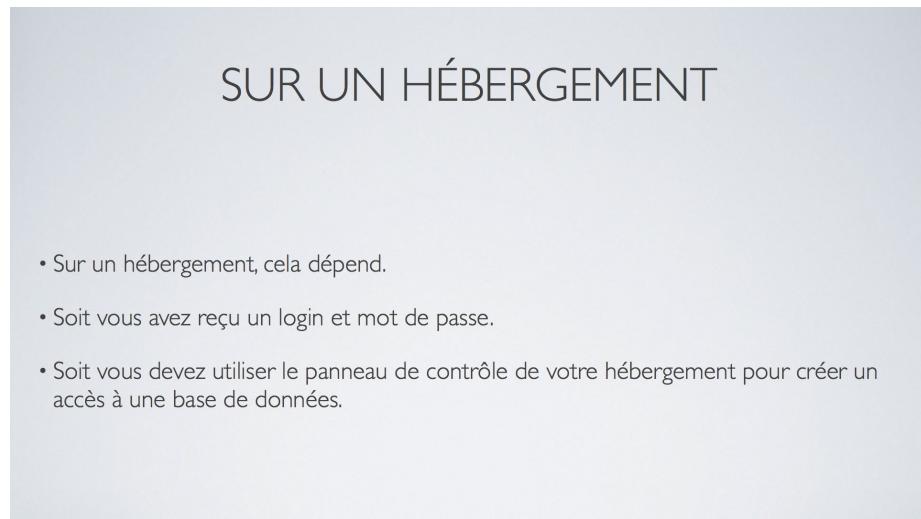
- «localhost» ou «127.0.0.1» est un moyen pour la machine de se désigner elle-même (et donc quelque soit son adresse IP).
- Lorsque l'on configure une application web en indiquant que l'adresse de la base de données est «localhost», cela signifie que le serveur de base de données se trouve sur la même machine que le serveur web (puisque c'est lui qui exécute les scripts php).

## SUR VOTRE ORDI

- Login MySql : root
- Password MySql sous Windoze : (il n'y en a pas)
- Password MySql sous MacOSX : root
- Adresse du serveur de base de données : localhost
- Préfixe des tables : spip

## SUR CARL

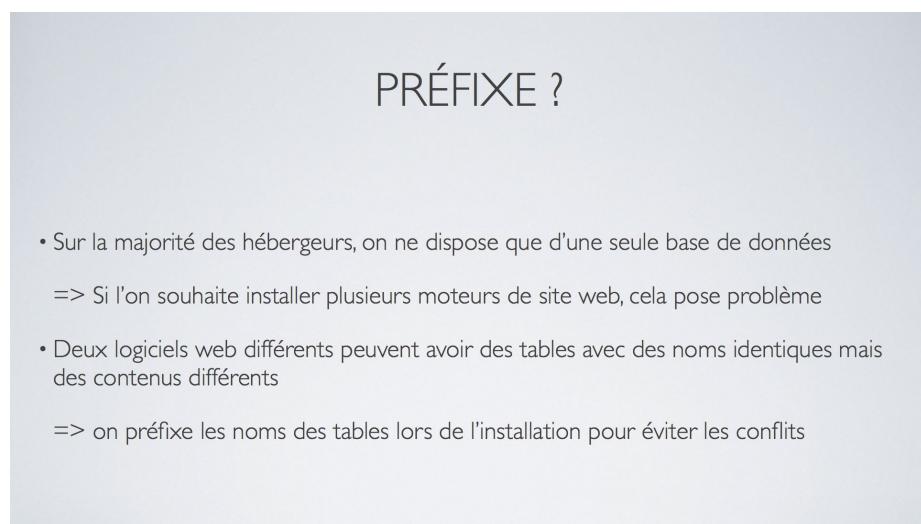
- Nom de la base de données : userXX
- Login MySql : userXX
- Password MySql : \*\*\*\* (le mot de passe fourni)
- Adresse du serveur de base de données : localhost
- Préfixe des tables : spip



## CHOIX DE LA BASE DE DONNÉES

The image shows a screenshot of a configuration interface for choosing a database. It includes two main sections:

- Choix de votre base :** A section where the user can select a database from a list or create a new one. The list includes "information\_schema", "mysql", and "performance\_schema". There is also an option to "Créer une nouvelle base de données:" with a text input field containing "spip".
- Préfixe des tables :** A section where the user can enter a prefix for database tables. The input field contains "spip".



## SPIP\_LOADER.PHP

- Copier le fichier spip\_loader dans un sous-dossier de «public\_html»
- Surfer sur le serveur et cliquer sur le fichier
- Ici pas besoin de régler les droits sur les dossiers «IMG,...», c'est spip\_loader qui le fait pour vous
- MAIS ...

## SPIP\_LOADER.PHP

Sur Carl uniquement !

- C'est le serveur HTTP qui télécharge et décomprime l'archive
  - => le propriétaire des fichiers est HTTP et pas vous (userXX)
  - => vous ne savez pas effacer les fichiers par FTP !

## SUPPRIMER UNE INSTALL SPIP\_LOADER.PHP

- Copier, par FTP, le fichier deltree.php (voir mon centre des ressources) à la racine du dossier spip
- Surfer sur <http://10.59.26.130/userXX/spip/deltree.php>



## Chapitre

# 6

### Sommaire

- 6.1 Documentation SPIP
- 6.2 Les balises propres au site
- 6.3 Syntaxe des boucles
- 6.4 Syntaxe complète des balises
- 6.5 Les autres pages
- 6.6 La balise #ENV
- 6.7 Les filtres
- 6.8 Secteur et branche
- 6.9 Une boucle dans une boucle
- 6.10 Compter le nombre d'élément d'une boucle
- 6.11 GET & SET
- 6.12 404, File not found
- 6.13 Faire un menu correctement
- 6.14 #EXPOSE
- 6.15 Boucle réusive
- 6.16 Les formulaires

## Conception de squelettes

Les deux notions fondamentales de SPIP permettant d'afficher des données sont les boucles et les balises. SPIP propose :

**des balises propres au site**, qui permettent d'afficher des données en provenance de l'interface d'administration. Par exemples : le nom du site, le slogan du site, URL du site, ...

**des boucles**, qui permettent de retrouver — de choisir — les rubriques, les articles, les brèves, ..., que l'on désire présenter.

**des balises propres aux boucles**, qui permettent d'afficher des données ( tel que le titre, le texte, ... ) à l'intérieur des boucles.

## 6.1 Documentation SPIP

### 6.1.1 Boucles et balises

Principe général	 <a href="http://www.spip.net/fr_article894.html">http://www.spip.net/fr_article894.html</a>
Syntaxe des boucles	 <a href="http://www.spip.net/fr_article898.html">http://www.spip.net/fr_article898.html</a>
Syntaxe des balises	 <a href="http://www.spip.net/fr_article899.html">http://www.spip.net/fr_article899.html</a>
Balises propres au site	 <a href="http://www.spip.net/fr_article1902.html">http://www.spip.net/fr_article1902.html</a>
La boucle RUBRIQUES	 <a href="http://www.spip.net/fr_article904.html">http://www.spip.net/fr_article904.html</a>
La boucle ARTICLES	 <a href="http://www.spip.net/fr_article902.html">http://www.spip.net/fr_article902.html</a>
La boucle BREVES	 <a href="http://www.spip.net/fr_article906.html">http://www.spip.net/fr_article906.html</a>

### 6.1.2 Critères et filtres

Critères communs	 <a href="http://www.spip.net/fr_article900.html">http://www.spip.net/fr_article900.html</a>
Filtres	 <a href="http://www.spip.net/fr_article901.html">http://www.spip.net/fr_article901.html</a>
Gestion des dates	 <a href="http://www.spip.net/fr_article1971.html">http://www.spip.net/fr_article1971.html</a>

### 6.1.3 D'autres pages

INCLURE d'autres squelettes	 <a href="http://www.spip.net/fr_article1828.html">http://www.spip.net/fr_article1828.html</a>
La boucle FORUMS	 <a href="http://www.spip.net/fr_article908.html">http://www.spip.net/fr_article908.html</a>
La boucle HIERARCHIE	 <a href="http://www.spip.net/fr_article913.html">http://www.spip.net/fr_article913.html</a>
La boucle MOTS	 <a href="http://www.spip.net/fr_article909.html">http://www.spip.net/fr_article909.html</a>
La gestion des pages 404	 <a href="http://www.spip.net/fr_article3226.html">http://www.spip.net/fr_article3226.html</a>
Le système de pagination	 <a href="http://www.spip.net/fr_article3367.html">http://www.spip.net/fr_article3367.html</a>

## 6.2 Les balises propres au site

Le contenu de ces balises est indépendant de tout contexte, leur contenu ne change que lorsque l'on modifie la page « configuration » dans l'espace privé de SPIP.

Voici une liste non-exhaustive, les balises de la première colonne portent des noms suffisamment explicites, les balises de la seconde colonne seront expliquées plus loin.

#NOM_SITE_SPIP	#DATE
#SLOGAN_SITE_SPIP	#REM
#URL_SITE_SPIP	#SELF
#DESCRIPTIF_SITE_SPIP	#URL_PAGE
#LOGO_SITE_SPIP	#CHEMIN
#CHARSET	#ENV
#LANG	...

## 6.3 Syntaxe des boucles

### 6.3.1 Boucle simple

De manière simpliste, on peut comparer une base de données à un ensemble de listes. Comme SPIP utilise une base de données pour stocker ces informations, on peut considérer qu'il manipule des listes d'articles, de rubriques, de brèves, ...

Les boucles permettent d'afficher l'entièreté ou une partie de ces listes. Leur syntaxe est comparable à du code HTML.

```
<BOUCLEx(TYPE)>
#BALISE
</BOUCLEx>
```

Le mot « BOUCLE » doit obligatoirement être écrit en majuscule, suivi d'un nom, suivi du type de la boucle. Le type doit également être écrit en majuscule.

Le nom de la boucle permet simplement de différencier les différentes boucles d'un fichier. Le nom d'une boucle doit donc être unique dans un fichier.

Le type de boucle correspond aux types de données ( aux types de listes ) gérées par SPIP. Le type se termine toujours par un « S ».

Les trois principaux types sont : RUBRIQUES, ARTICLES, BREVES.

L'intérieur de la boucle peut contenir des balises propres à la boucle ou au site, mais également du code HTML.

Dans l'exemple suivant, je considère que la liste « RUBRIQUES » contient 3 éléments.

```
<ul>
<li> avant la boucle </li>

<BOUCLE_rub(RUBRIQUES)>
<li> #BALISE </li>
</BOUCLE_rub>

<li> après la boucle </li>
</ul>
```

Le code ci-contre donne le résultat suivant :

```
<ul>
<li> avant la boucle </li>
<li> ... </li>
<li> ... </li>
<li> ... </li>
<li> après la boucle </li>
</ul>
```

Pour connaître les balises disponibles pour chaque type de boucle, *il faut consulter la documentation*.

Si la liste des rubriques contenait 100 éléments, la boucle précédente affichera les 100 éléments. Il ne manque plus qu'un moyen de « choisir » les éléments à afficher. Ce moyen se nomme critère.



On retiendra qu'une boucle sans critère permet d'afficher tous les éléments, du type de la boucle, publiés sur le site.

### 6.3.2 Boucle avec critères

Les critères sont des mots écrits en minuscule placés entre accolades. Certains sont communs à toutes les boucles, d'autres sont spécifiques à une boucle. Une boucle peut contenir plusieurs critères.

Il y a deux types de critères,

- ceux qui permettent de choisir les éléments à afficher,
- ceux qui permettent de trier l'ordre d'affichage.

La syntaxe est la suivante :

```
<BOUCLEx(TYPE){critère1}...{critèreN}>
#BALISE
</BOUCLEx>
```

Le critère « `racine` » permet de n'afficher que les rubriques à la racine du site.

```
<BOUCLE_1(RUBRIQUES){racine}>
#TITRE
</BOUCLE1>
```

Le critère « `par date` » permet de trier le résultat de la boucle du plus vieux au plus récent.

```
<BOUCLE_2(RUBRIQUES){par date}>
#TITRE #DATE <br />
</BOUCLE_2>
```

Le critère « `inverse` » permet d'inverser l'ordre de tri, on obtient ainsi un tri allant du plus jeune au plus ancien.

```
<BOUCLE_3(RUBRIQUES){par date}{inverse}>
#TITRE #DATE <br />
</BOUCLE_3>
```

Le critère `{a,b}` permet de ne choisir que certains éléments du résultat. Il faut voir le résultat de la boucle comme une liste numérotée à partir de 0. Le nombre total d'items de la liste est noté `n`.

- `a` indique l'indice à partir duquel on désire commencer l'affichage.
- `b` indique le nombre d'items à afficher.

La boucle suivante affichera l'article le plus ancien.

```
<BOUCLE_4(RUBRIQUES){par date}{0,1}>
#TITRE #DATE
</BOUCLE_4>
```

La boucle suivante affichera l'article le plus récent, c'est à dire le dernier item de la liste. Si la liste comporte 10 items, `n` vaut 10. Comme la liste est numérotée de 0 à 9, pour sélectionner le dernier item, « `a` » doit valoir 9, soit `n-1`.

```
<BOUCLE_5(RUBRIQUES){par date}{n-1,1}>
#TITRE #DATE
</BOUCLE_5>
```

La boucle suivante affiche également l'item le plus récent.

```
<BOUCLE_6(RUBRIQUES){par date}{inverse}{0,1}>
#TITRE #DATE <br />
</BOUCLE_6>
```

Comparer le résultat de ces deux boucles.

```
<BOUCLE_7(RUBRIQUES){par date}{inverse}{0,5}>
#TITRE #DATE <br />
</BOUCLE_7>

<BOUCLE_8(RUBRIQUES){par date}{n-5,5}>
#TITRE #DATE <br />
</BOUCLE_8>
```

La première boucle trie les éléments du plus jeune au plus ancien. La seconde boucle trie du plus ancien au plus jeune. Ces deux boucles affichent toutes les deux les 5 rubriques les plus récentes, la différence est l'ordre d'affichage.

### 6.3.3 Syntaxe complète des boucles

Il peut arriver qu'une boucle ne renvoie aucun résultat. Il peut alors être utile d'afficher un message en conséquence.

```
<B_9>
<ul>
<BOUCLE_9(BREVES)>
<li> #TITRE </li>
</BOUCLE_9>
</ul>

</B_9>
<h2> pas de brèves </h2>
</B_9>
```

Si des brèves sont publiées, la boucle donnera :

```
<ul>
<li> brève 1</li>
...
</ul>
```

Sinon :

```
<h2> pas de brèves </h2>
```

## 6.4 Syntaxe complète des balises

Le code suivant permet d'afficher le nom et le slogan du site. Si le champ slogan est resté vide dans l'interface d'administration, le code ci-dessous

```
<h1 style="background-color:red"> #NOM_SITE_SPIP </h1>
<h2 style="background-color:green; height:10px">
    #SLOGAN_SITE_SPIP
</h2>
```

donnerait comme résultat le code HTML suivant :

```
<h1 style="background-color:red"> Intro R&B </h1>
<h2 style="background-color:green; height:10px"> </h2>
```

C'est à dire graphiquement :



La syntaxe complète des balises permet de supprimer du contenu dans le cas où une balise est vide.

```
[ code html optionnel avant (#BALISE) code html optionnel après ]
```

Ce qui donne avec l'exemple ci-dessus :

```
<h1 style="background-color:red">#NOM_SITE_SPIP</h1>
[ <h2 style="background-color:green; height:10px">
    (#SLOGAN_SITE_SPIP)
</h2> ]
```

Le code HTML généré est donc

```
<h1 style="background-color:red">Intro R&B</h1>
```

## 6.5 Les autres pages

Chaque boucle propose une balise du type #URL\_xxxx. On a ainsi :

- #URL\_RUBRIQUE dans la boucle rubrique,
- #URL\_ARTICLE dans la boucle article,
- #URL\_BREVE dans la boucle brève,
- ...

Ces balises génèrent des URL de la forme .../spip.php?page=xxxx&id\_xxxx=y.

- xxxx vaut « rubrique » pour la boucle RUBRIQUES, « article » pour la boucle ARTICLES, ...
- « y » est un numéro de rubrique, d'article, ...

Par exemples : `.../spip.php?page=rubrique&id_rubrique=1`.

- « spip.php » est le script php principal de SPIP.
- « ? » permet de séparer la partie adresse et la partie paramètres de l'URL.
- La syntaxe des paramètres est : `param=valeur`, chaque paramètre étant séparé par un &.
- Le paramètre « page » indique à SPIP d'utiliser un squelette nommé rubrique.html.
- Le paramètre « id\_rubrique » permet d'indiquer le numéro de la rubrique sur laquelle on souhaite travailler.



Chaque type de boucle propose une balise #URL\_xxx. On peut donc créer un squelette par type de boucle. A vous de choisir ce que chaque squelette affiche.

Les boucles peuvent utiliser les paramètres de l'URL comme valeur pour les critères. Dans l'exemple suivant, le critère « id\_rubrique » prend la valeur « 5 ».

La première boucle affiche le titre et le texte de la rubrique 5, la seconde boucle affiche la liste des articles se trouvant dans la rubrique 5.

```
URL : .../spip.php?page=rubrique&id_rubrique=5

<BOUCLE_10(RUBRIQUES){id_rubrique}>
  #TITRE <br />
  #TEXTE <hr />
</BOUCLE_10>

<BOUCLE_11(ARTICLES){id_rubrique}>
  #TITRE <br />
  #TEXTE <hr />
</BOUCLE_11>
```

La boucle suivante affiche l'article numéro 5.

```
URL : .../spip.php?page=article&id_article=5

<BOUCLE_12(ARTICLES){id_article}>
  #TITRE <br />
  #TEXTE <hr />
</BOUCLE_12>
```

Une boucle utilisant un paramètre de l'URL, ne sait pas être calculée si le paramètre n'est pas dans l'URL. La boucle suivante n'affiche donc rien.

```
URL : .../spip.php?page=article

<BOUCLE_13(ARTICLES){id_article}>
  #TITRE <br />
  #TEXTE <hr />
</BOUCLE_13>
```

## 6.6 La balise #ENV

La balise #ENV permet de récupérer la valeur d'un paramètre de l'URL. Le code suivant, placé dans la page article.html, affiche « 5 » comme résultat.

```
URL : .../spip.php?page=article&id_article=5
[ (#ENV{id_article}) ]
```

Il est également possible de passer vos propres paramètres sur l'URL, ceux-ci doivent être ajoutés lorsque vous fabriquez les liens.

```
<!-- fichier sommaire -->
<BOUCLE_14(RUBRIQUES){racine}>

<a href="#URL_RUBRIQUE&var=zorro"> #TITRE </a>
</BOUCLE_14>

<!-- fichier rubrique -->
<h2>Teste de la balise #ENV</h2>

[ var = (#ENV{var}) ]
```

Lors du clic sur un lien de la page sommaire, l'URL générée sera par exemple : [.../spip.php?page=rubrique&id\\_rubrique=2&var=zorro](#).

La page rubrique affichera alors ceci :

```
Teste de la balise #ENV
var = zorro
```

Le contenu de « var » peut également provenir d'une balise SPIP.

```
<BOUCLE_15(RUBRIQUES){racine}>

<a href="#URL_RUBRIQUE&var=#ID_SECTEUR"> #TITRE </a>
</BOUCLE_15>
```

## 6.7 Les filtres

Les filtres portent uniquement sur les balises. Pour utiliser un filtre, il faut obligatoirement utiliser la syntaxe complète des balises « [(#BALISE)] ».

### 6.7.1 Filtres sur les dates

Par défaut les dates sont affichées sous la forme YYYY-MM-DD hh:mm:ss, les filtres permettent d'arranger cela. Lisez [http://www.spip.net/fr\\_article1971.html](http://www.spip.net/fr_article1971.html) pour plus d'informations.

```
[(#DATE|affdate)]
[(#DATE|nom_jour)], le [(#DATE|jour)] [(#DATE|nom_mois)]
```

### 6.7.2 Le retour de if else

Il faudra ici utiliser deux filtres l'un à la suite de l'autre.

Les filtres suivants permettent de tester des égalités, des différences, ... La valeur de la balise est comparée avec le contenu des accolades.

```
[(#BALISE|=={})]  [(#BALISE|!={})]
[(#BALISE|>{})]  [(#BALISE|<={})]
[(#BALISE|>{})]  [(#BALISE|<{})]
```

Seuls, ces filtres ne sont pas utiles, il faut utiliser le filtre « ?{oui,non} ».

```
[(#BALISE|=={a}|?{oui,non})]
```

- Si le contenu de #BALISE vaut « a », alors le « oui » sera affiché,
- sinon, le mot « non » sera affiché.

Ceci peut être utilisé pour ajouter un attribut « class » à un tag HTML en fonction du contenu d'une balise.

La boucle suivante est codée dans le fichier sommaire, elle permet de réaliser un menu des rubriques à la racine du site. Sur la page sommaire, on est toujours sur l'onglet « Home ». C'est donc toujours l'onglet « Home » qui sera actif.

```
<ul>
  <li class="selected"><a href="#URL_SITE_SPIP">Home</a></li>

  <BOUCLE_menu(RUBRIQUES){racine}>
    <li><a href="#">#URL\_RUBRIQUE"> #TITRE </a></li>
  </BOUCLE\_menu>
</ul>
```

Le code HTML généré sur la page sommaire est le suivant :

```
<ul>
  <li class="selected"><a href="#URL_SITE_SPIP">Home</a></li>
  <li><a href="#">.../spip.php?page=rubrique&id\_rubrique=1"> Rub1 </a></li>
  <li><a href="#">.../spip.php?page=rubrique&id\\_rubrique=2"> Rub2 </a></li>
  <li><a href="#">.../spip.php?page=rubrique&id\\\_rubrique=3"> Rub3 </a></li>
</ul>
```

La boucle suivante est codée dans le fichier rubrique.html. Lors de l'appel à la page rubrique, l'URL est par exemple : `.../spip.php?page=rubrique&id_rubrique=2`.

```
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>

<BOUCLE_menu(RUBRIQUES){racine}>
<li [(#ID_RUBRIQUE|=={2}|?{class="selected", ''})]>
  <a href="#URL_RUBRIQUE"> #TITRE </a>
</li>
</BOUCLE_menu>
</ul>
```

Ce code permet d'ajouter « `class="selected"` » dans le `<li>` uniquement lorsque `#ID_RUBRIQUE` vaut 2. Point de vue HTML, cela donne :

```
<ul>
<li><a href=".../spip.php">Home</a></li>
<li><a href=".../spip.php?page=rubrique&id_rubrique=1">
  Rub1 </a></li>
<li class="selected"><a
  href=".../spip.php?page=rubrique&id_rubrique=2"> Rub2
</a></li>
<li><a href=".../spip.php?page=rubrique&id_rubrique=3">
  Rub3 </a></li>
</ul>
```

Afin de ne pas devoir créer un fichier rubrique.html pour chaque rubrique, il faut un moyen de remplacer le {2}.

Si on utilise la balise `#ENV{id_rubrique}` dans le fichier rubrique.html, sa valeur ne change que lorsque l'on clique sur le menu et que l'on demande à SPIP de calculer une nouvelle page rubrique.html.

Pour s'en convaincre, examinons le code suivant à placer dans le fichier rubrique.html.

```
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>

<BOUCLE_menu(RUBRIQUES){racine}>
<li><a href="#URL_RUBRIQUE">
  #TITRE - #ID_RUBRIQUE - #ENV{id_rubrique}

</a></li>
</BOUCLE_menu>
</ul>
```

Si on clique sur lien pointant vers la rubrique 2, le code HTML généré est :

```
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>
<li><a href=".../spip.php?page=rubrique&id_rubrique=1">
    Rub1 - 1 - 2 </a></li>
<li><a href=".../spip.php?page=rubrique&id_rubrique=2">
    Rub2 - 2 - 2 </a></li>
<li><a href=".../spip.php?page=rubrique&id_rubrique=3">
    Rub3 - 3 - 2 </a></li>
</BOUCLE_menu>
</ul>
```

Le code suivant permet d'avoir une seule page rubrique.html qui active le bon onglet de menu quelque soit la rubrique demandée.

```
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>

<BOUCLE_menu(RUBRIQUES){racine}>
<li [ (#ID_RUBRIQUE|=={#ENV{id_rubrique}})|?{class="selected", ''}]>
    <a href="#URL_RUBRIQUE"> #TITRE </a>
</li>
</BOUCLE_menu>
</ul>
```

Il reste un problème, si l'on est descendu dans une sous-rubrique, #ENV{id\_rubrique} ne sera jamais plus égale à aucun #ID\_RUBRIQUE de notre menu. Notre système ne fonctionne donc plus.

## 6.8 Secteur et branche

Considérons l'arborescence de rubriques suivante :

```
\label{arbo}
Racine
|
|---Rub1---Rub1.1---Rub1.1.1
|   |
|   |   |
|   |   |---Rub1.1.2
|   |
|   |   |
|   |   |---Rub1.2
|
|---Rub2---Rub2.1
|   |
|   |---Rub2.2
```

Chaque rubrique porte un numéro, « l'id de rubrique ». Ce qui donne par exemple :

Nom Rubrique	id_rubrique	Nom Rubrique	id_rubrique
Rub1	1	Rub2.1	5
Rub2	2	Rub1.2	6
Rub1.1	3	Rub2.2	7
Rub1.1.1	4	Rub1.1.2	8

Ces numéros sont attribués dans l'ordre de création des rubriques.

L'id de *secteur* d'une rubrique correspond à l'id de la rubrique parent se trouvant à la racine. Dans notre exemple, pour Rub1.1, Rub1.2, Rub1.1.1 et Rub1.1.2, l'id de secteur correspond à l'id de la rubrique Rub1, soit 1. De même pour Rub2.1 et Rub2.2, l'id secteur correspond à l'id de la rubrique Rub2, soit 2.

Comme les rubriques racines sont le point de départ des secteurs, on peut considérer qu'elles n'ont pas d'id de secteur. Plus exactement, il vaut 0, c'est à dire la valeur de la racine.

On parle de branche lorsque l'on considère l'ensemble des rubriques d'un secteur.

## 6.9 Une boucle dans une boucle

Lorsqu'une boucle s'exécute dans un squelette, la valeur des critères sont pris sur l'URL. Lorsqu'une boucle s'exécute dans une autre boucle, la valeur des critères sont pris sur les balises de la boucle englobante (c'est automatique, il n'y a rien à faire).

```
<BOUCLE_1(RUBRIQUES){id_rubrique}>
#TITRE
<BOUCLE_2(RUBRIQUES){id_parent}>
#TITRE
</BOUCLE_2>
</BOUCLE_1>
```

La balise #TITRE de la boucle 1 permet d'afficher le titre de la rubrique dont le numéro est id\_rubrique. Si id\_rubrique vaut 3, #TITRE affichera Rub1.1.

Le critère id\_parent de la boucle 2 permet de sélectionner la liste des rubriques contenues dans la rubrique courante. La balise #TITRE de cette boucle affichera donc les titres des sous-rubriques de Rub1.1, soit Rub1.1.1 et Rub1.1.2.

Si le critère de la seconde boucle devient id\_enfant, elle affiche le titre de la rubrique qui a pour enfant une rubrique dont le numéro est l'#ID\_RUBRIQUE de la boucle 1. C'est à dire, Rub1.

```
<BOUCLE_1(RUBRIQUES){id_rubrique}>
#TITRE
<BOUCLE_2(RUBRIQUES){id_enfant}>
#TITRE
</BOUCLE_2>
</BOUCLE_1>
```

Lorsque l'on écrit des boucles imbriquées, il ne faut pas perdre de vue que la boucle interne va s'exécuter à chaque tour effectué dans la boucle externe. Prenons l'exemple suivant :

```
<BOUCLE_1(RUBRIQUES){racine}>
#TITRE
<BOUCLE_2(RUBRIQUES){id_parent}>
#TITRE
</BOUCLE_2>
</BOUCLE_1>
```

La boucle 1 fera deux tours et affichera Rub1 et Rub2. Lors du premier tour de BOUCLE\_1, la BOUCLE\_2 affichera Rub1.1 et Rub1.2. Lors du second tour de BOUCLE\_1, la BOUCLE\_2 affichera Rub2.1 et Rub2.2.

Ceci signifie que l'environnement (la valeur des critères) de la BOUCLE\_1 ne change pas, tandis que l'environnement de la BOUCLE\_2 change à chaque tour dans la boucle 1.

On dit que la BOUCLE\_2 s'exécute dans le contexte de la BOUCLE\_1.

## 6.10 Compter le nombre d'élément d'une boucle

Il est possible de compter le nombre de résultat trouvé par une boucle avec la balise #TOTAL\_BOUCLE. Cette balise peut être utilisée n'importe où dans la boucle, même dans la partie alternative de la boucle.

La balise #COMPTEUR\_BOUCLE permet de connaître l'itération actuelle de la boucle.

Pour connaître le nombre d'éléments d'une boucle sans rien afficher, on utilisera par exemple :

```
<BOUCLE_a(ARTICLES)> </BOUCLE_a>#TOTAL_BOUCLE</B_a>
```

La notation racourcie suivante est également possible :

```
<BOUCLE_a(ARTICLES) />#TOTAL_BOUCLE</B_a>
```

## 6.11 GET & SET

La balise #SET va permettre de sauvegarder une donnée. La balise #GET, de la récupérer.

Ceci peut servir dans le cas où l'on souhaite mettre un 's' à un mot en fonction du nombre d'éléments d'une boucle. Par exemple pour des commentaires :

```
<!-- squelette article.html -->
<BOUCLE_nbrcom(FORUMS){id_article}>
  #SET{nbrCom,#TOTAL_BOUCLE}
</B_nbrcom>

...
<p> Il y a #GET{nbrCom}
  [(#GET{nbrCom}|<={1}|?{commentaire,commentaires})]</p>
```

## 6.12 404, File not found

Lorsque l'on demande à un serveur web un fichier (une page) qu'il ne possède pas, celui-ci retourne un message d'erreur du type : « Error 404, file not found ».

Pour SPIP, on appelle toujours le script « spip.php ». Le serveur web n'enverra donc jamais d'erreur 404.

Considérons le squelette article.html suivant :

```

<div id="menu">
<BOUCLE_menu(RUBRIQUES){racine}>
#TITRE
</BOUCLE_menu>
</div>

<div id="sidebar">
<BOUCLE_news(BREVES){par date}{inverse}{0,5}>
#TITRE
</BOUCLE_news>
</div>

<div id="content">
<BOUCLE_content(ARTICLES){id_article}>
<h3>TITRE</h3>
#TEXTE
</BOUCLE_content>
</div>

```

Si vous demandez à SPIP d'afficher un article qui n'existe pas (ou qui n'est plus publié) en accédant directement à celui-ci par son URL ([.../spip.php?page=article&id\\_article=127](#)), SPIP vous affichera le menu et la sidebar, le content, lui, restera vide.

Bref, frustration pour l'internaute qui n'a pas de contenu, ni de message pour lui indiquer que l'article demandé n'existe pas.

Dans le cas où un fichier squelette n'affiche absolument rien, SPIP affiche un message d'erreur 404.

Pour tester ce mécanisme, videz complètement votre fichier article.html. A l'URL de base de votre site SPIP, ajoutez : « [.../spip.php?page=article&id\\_article=12700](#) ». SPIP vous affichera son message 404 par défaut.

Nous allons réécrire notre squelette article.html, la boucle « content » va maintenant englober complètement le fichier html.

```

<BOUCLE_content(ARTICLES){id_article}>

<div id="menu">
<BOUCLE_menu(RUBRIQUES){racine}>
#TITRE
</BOUCLE_menu>
</div>

<div id="sidebar">
<BOUCLE_news(BREVES){par date}{inverse}{0,5}>
#TITRE
</BOUCLE_news>
</div>

<div id="content">
<h3>TITRE</h3>
#TEXTE
</div>

</BOUCLE_content>

```

Si l'article demandé n'existe pas, la boucle « content » ne retourne rien. Le squelette est donc vide et spip affiche son message 404.

Vous noterez que maintenant, les autres boucles (menu et news) ne prennent plus leurs paramètres sur l'URL mais dans la boucle « content » (ce qui ne change rien ici).

Pour personnaliser le message 404, il suffit de créer un fichier nommé « 404.html ». Le plus simple est de copier le fichier sommaire.html vers 404.html. Dans la section « content » du fichier 404.html, on placera un message explicite.

Voici un exemple de fichier 404.html.

```
<!DOCTYPE html>
...
<div id="menu">
<BOUCLE_menu(RUBRIQUES){racine}>
#TITRE
</BOUCLE_menu>
</div>

<div id="sidebar">
<BOUCLE_news(BREVES){par date}{inverse}{0,5}>
#TITRE
</BOUCLE_news>
</div>

<div id="content">
<p>
La page demandée n'existe pas !
<a href="#URL_SITE_SPIP">Home</a>
</p>

...
</div>
...
```



Vous noterez que la totalité du fichier, <!DOCTYPE html> y compris, doit être dans la boucle « content ».



Cette technique peut être utilisée dans tous les fichiers qui reçoivent des paramètres via l'url. Les pages sommaire.html et 404.html ne recevant jamais de paramètres, elles n'utiliseront jamais cette astuce.



La boucle de gestion de l'erreur 404, que je nommerai toujours « BOUCLE\_404 » doit être du même type que la page.

La mise en oeuvre de la page 404.html n'est pas obligatoire pour afficher un message d'erreur. Nous aurions également pu utiliser la syntaxe complète de la boucle comme suit :

```
<!DOCTYPE html>
...
<div id="menu">
<BOUCLE_menu(RUBRIQUES){racine}>
#TITRE
</BOUCLE_menu>
</div>

<div id="sidebar">
<BOUCLE_news(BREVES){par date}{inverse}{0,5}>
#TITRE
</BOUCLE_news>
```

```
</div>

<div id="content">
  <B_content>
    <BOUCLE_content(ARTICLES){id_article}>
      #TITRE ...
    </BOUCLE_content>
  </B_content>
  
</B_content>
</div>
...
```

Comment choisir entre ces deux techniques ?

- Si les autres boucles du squelette sont indépendantes du contexte, la technique de la page 404 ne se justifie pas.
- Par contre, si les autres boucles du fichier sont dépendantes du contexte, par exemple avec un critère id\_parent, alors il faut utiliser la technique de la page 404. En effet, celles-ci n'afficheront plus les données que vous souhaitiez ou n'afficheront plus rien. La page 404.html permet donc de garder une page cohérente.
- L'autre avantage d'avoir une boucle qui englobe la totalité du fichier est que l'on peut utiliser les données de cette boucle à tout endroit du fichier comme dans la section <head> par exemple.

## 6.13 Faire un menu correctement

### 6.13.1 Activer le bon lien

Un menu simple est d'afficher la liste des rubriques principales du site. Il s'agit des « dossiers » situés à la racine du site d'où le nom du critère.

Généralement, le lien de menu actif se démarque des autres par un style css différent, ici `class="selected"`.

Sur la page sommaire.html, c'est toujours le lien « Home » qui est actif.

```
<!-- Page sommaire.html -->
<ul>

  <li class="selected"> <a href="#URL_SITE_SPIP">Home</a> </li>

  <BOUCLE_menu(RUBRIQUES){racine}>
    <li> <a href="#URL_RUBRIQUE"> #TITRE </a> </li>
  </BOUCLE_menu>
</ul>
```

Sur les autres pages, ce n'est plus jamais le lien home qui est actif, on peut donc écrire ceci :

```
<!-- Page rubrique.html -->
<ul>
  <li><a href="#URL_SITE_SPIP">Home</a></li>
  <BOUCLE_menu(RUBRIQUES){racine}>
```

```
<li class="selected"> <a href="#URL_RUBRIQUE" #TITRE </a> </li>
</BOUCLE_menu>
</ul>
```

Le problème c'est que tous les `<li>` dans la boucle recoivent `class="selected"`.

Il faut donc un moyen d'ajouter `class="selected"` uniquement lorsque la boucle écrit la ligne `<li>` correspondant à la rubrique affichée.

L'id de rubrique actuellement affiché se trouve sur l'URL, la balise `#ENV` va nous permettre de le récupérer.

Nous allons utiliser un filtre qui compare l'id de rubrique que la boucle ajoute au menu, à celui de l'url.

```
<!-- Page rubrique.html -->
<ul>
  <li><a href="#URL_SITE_SPIP">Home</a></li>
  <BOUCLE_menu(RUBRIQUES){racine}>

  <li [(#ID_RUBRIQUE =={#ENV{id_rubrique}}) | ?{class="selected", ''}]>
    <a href="#URL_RUBRIQUE&param=#ID_SECTEUR"> #TITRE </a>
  </li>
  </BOUCLE_menu>
</ul>
```

Si maintenant la rubrique affichée est une sous-rubrique, cela ne marche plus ! Reprenons l'arborescence définie plus haut ??, les id de rubrique à la racine sont : 1 et 2. Si nous affichons la rubrique « Rub2.2 », dont l' id de rubrique vaut 7, la condition du filtre ne sera jamais vrai et donc `class="selected"` ne sera jamais ajouté.

Grâce à la boucle de gestion de l'erreur 404 et du couple de balises `#SET` et `#GET`, nous allons utiliser l'id de secteur de la rubrique actuellement affiché. Pour rappel l'id de secteur d'un objet SPIP correspond à l'id de rubrique parent se trouvant à la racine du site.

```
<!-- Page rubrique.html -->
<BOUCLE_404(RUBRIQUES){id_rubrique}>

#SET{idsecteur,#ID_SECTEUR}

<!DOCTYPE html>
...
<ul>
  <li><a href="#URL_SITE_SPIP">Home</a></li>
  <BOUCLE_menu(RUBRIQUES){racine}>

  <li [ (#ID_RUBRIQUE =={#GET{idsecteur}}) | ?{class="selected", ''}]>
    <a href="#URL_RUBRIQUE" #TITRE </a>
  </li>
  </BOUCLE_menu>
</ul>
...
</BOUCLE_404>
```

Ceci est lourd et de plus les développeurs de SPIP ont prévu une balise qui fait la même chose. On peut donc simplifier comme ceci :

```
<!-- Page rubrique.html -->
<!DOCTYPE html>
...
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>
<BOUCLE_menu(RUBRIQUES){racine}>

<li [(#EXPOSE{class="selected"})]>    <a href="#URL_RUBRIQUE">
    #TITRE </a> </li>

</BOUCLE_menu>
</ul>
...
```

### 6.13.2 Trier les liens

La méthode la plus simple pour trier les liens d'un menu est de les numérotter. Il vous suffit donc d'aller dans l'espace privé et de numérotter les rubriques racines.



Attention le numéro doit être suivi d'un point et d'un espace : « 1. Titre-rubrique-un. »

Il n'y a plus qu'à ajouter le critère de tri « {par numero} » à notre menu :

```
<!-- Page rubrique.html -->
...
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>

<BOUCLE_menu(RUBRIQUES){racine}{par numero}>

<li [(#EXPOSE{class="selected"})]>    <a href="#URL_RUBRIQUE">
    #TITRE </a> </li>
</BOUCLE_menu>
</ul>
...
```



Si vous ne faites pas d'espace entre le numéro et le titre : « 1.Titre-rubrique-un. », il faudra appliquer le filtre « !supprimer\_numero » à la balise #TITRE : [(#TITRE|supprimer\_numero)]

### 6.13.3 Pages statiques dans le menu

Est-il raisonnable de créer une rubrique « Contact » ou « A propos » juste pour y placer une page ?

On peut décider de « hard coder » ces deux pages et de les placer dans le dossier « squelettes ». Il ne faut pas perdre de vue que seul le webmaster sera capable de modifier ces deux pages !

Le code du menu peut être alors :

```
<!-- Page rubrique.html -->
...
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>
```

```
<BOUCLE_menu(RUBRIQUES){racine}{par numero}>
<li [(#EXPOSE{class="selected"})]>
  <a href="#URL_RUBRIQUE> #TITRE </a>
</li>

<li> <a href="#URL_PAGE{contact.html}">Contact</a> <li>
<li> <a href="#URL_PAGE{about.html}">A propos</a> <li>

</BOUCLE_menu>
</ul>
...

```

On peut également créer une rubrique nommée par exemple « 0. Statiques » et y placer les deux articles « Contact » et « A propos ». Il suffit de ne jamais afficher la rubrique 0 dans le menu. Il faut ensuite afficher les deux articles dans le menu (dans l'exemple je considère que l'id\_rubrique de la rubrique « 0. Statiques » vaut 1).

```
<!-- Page rubrique.html -->
...
<ul>
<li><a href="#URL_SITE_SPIP">Home</a></li>

<BOUCLE_menu1(RUBRIQUES){racine}{par numero}{1,n}>

<li [(#EXPOSE{class="selected"})]>
  <a href="#URL_RUBRIQUE> #TITRE </a>
</li>
</BOUCLE_menu1>

<BOUCLE_menu2(ARTICLES){id_rubrique=1}{par numero}>

<li [(#EXPOSE{class="selected"})]>
  <a href="#URL_ARTICLE> #TITRE </a>
</li>

</BOUCLE_menu2>
</ul>
...

```

## 6.14 #EXPOSE

La balise `#EXPOSE{oui,non}` (introduite dans la section précédente) affiche « oui » si l'objet SPIP est le principal de la page, sinon elle affiche « non ».

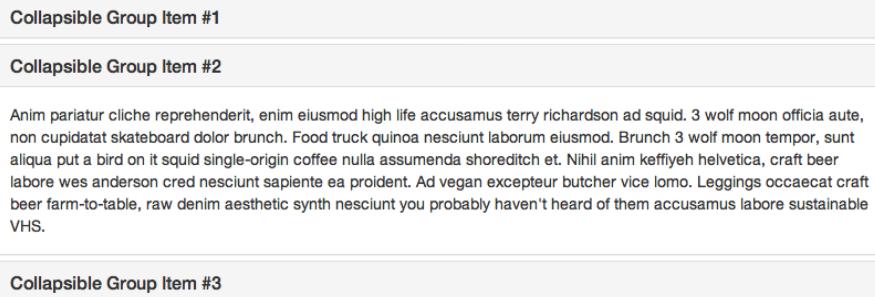
Dans l'exemple suivant on souhaite afficher la liste des articles d'une rubrique par ordre chronologique et mettre en évidence l'article dont l'id est sur l'url.

la balise `#EXPOSE` permet d'appliquer le style « collapse » sur tous les articles. Pour l'article principal (celui dont l'id est dans url), le style est « collapse-in ».

```
<!-- Page article.html -->
<BOUCLE_404(ARTICLES){id_article}>
<!DOCTYPE html>
...
<BOUCLE_content(ARTICLE){id_rubrique}{par date}>
<h2><a href="#URL_ARTICLE> #TITRE </a></h2>
```

```
<div [class="#EXPOSE{collapse-in,collapse}"]> #TEXTE </div>
</BOUCLE_content>
...
</BOUCLE_404>
```

Ce qui donne par exemple :



## 6.15 Boucle récursive

Cet type de boucle s'applique très bien à l'affichage des commentaires repris sous un article.

Voyons d'abord comment faire sans boucle récursive et pourquoi elle est nécessaire. Le code suivant affiche les commentaires d'un article et les réponses aux commentaires.

```
<BOUCLE_commentaires(FORUMS){id_article}>
...
<BOUCLE_reponse(FORUMS){id_parent}>
...
</BOUCLE_reponse>
...
</BOUCLE_commentaires>
```

Si l'on souhaite afficher la réponse à un réponse, le code devient :

```
<BOUCLE_commentaires(FORUMS){id_article}>
...
<BOUCLE_reponse(FORUMS){id_parent}>
...
<BOUCLE_reponse2(FORUMS){id_parent}>
...
</BOUCLE_reponse2>
...
</BOUCLE_reponse>
...
</BOUCLE_commentaires>
```

On comprend vite qu'il n'est pas possible de s'en sortir de cette manière. On ne va quand même pas écrire dix boucles « reponseX » imbriquées l'une dans l'autre au cas où !

La syntaxe récursive est particulière, la voici :

```
<BOUCLE_1...>
...
<BOUCLE_2(BOUCLE_1)></BOUCLE_2>
...
</BOUCLE_1>
```

La boucle « 1 » affiche ses données et appelle la boucle « 2 », qui elle reprend le même schéma que la boucle « 1 » et ainsi de suite.

Avec le code html et css nécessaire, cela peut donner ceci :

## Commentaires

Prototype du lanceur - le 16 octobre 2014, par Frédéric Gerard Répondre

commentaire 1

Prototype du lanceur - le 16 octobre 2014, par Frédéric Gerard Répondre

commentaire 2

Prototype du lanceur - le 16 octobre 2014, par Frédéric Gerard Répondre

commentaire 5

Prototype du lanceur - le 16 octobre 2014, par Frédéric Gerard Répondre

commentaire 3

Prototype du lanceur - le 16 octobre 2014, par Frédéric Gerard Répondre

commentaire 4

Dans le cas de la boucle FORUMS, il faut procéder comme suit :

1. écrire la boucle FORUMS qui retrouve les commentaires de l'article ;
2. écrire une boucle qui retrouve les réponses aux commentaires ;
3. écrire la boucle récursive.

Ce qui donne :

```
<BOUCLE_comm(FORUMS){id_article}>
...
<BOUCLE_comm2(FORUMS){id_parent}>
...
<BOUCLE_comm3(BOUCLE_comm2)></BOUCLE_comm3>
...
</BOUCLE_comm2>
...
```

```
</BOUCLE_comm>
</B_comm>
```

## 6.16 Les formulaires

SPIP met à notre disposition plusieurs formulaires, ceux-ci s'affichent avec de simples balises :

— #FORMULAIRE_RECHERCHE, — #FORMULAIRE_FORUM, — #FORMULAIRE_SIGNATURE, — #FORMULAIRE_SITE, — #FORMULAIRE_ECRIRE_AUTEUR,	— #FORMULAIRE_INSCRIPTION, — #LOGIN_PRIVE, — #LOGIN_PUBLIC, — #URL_LOGOUT,
---	---

La description de ces balises se trouve à cette url : [http://www.spip.net/fr\\_article1827.html](http://www.spip.net/fr_article1827.html).

Pour rappel, un formulaire HTML possède un attribut « action » qui permet d'indiquer l'adresse d'un script qui traitera le formulaire.

### 6.16.1 Formulaire de recherche

La documentation décrit assez bien le fonctionnement du moteur de recherche de SPIP [http://www.spip.net/fr\\_rubrique471.html](http://www.spip.net/fr_rubrique471.html).

La balise #FORMULAIRE\_RECHERCHE redirige, par défaut, vers le squelette recherche. Il est donc nécessaire de créer cette page.

Si vous souhaitez afficher le résultat dans une autre page, il faut indiquer l'url de la page entre {} : #FORMULAIRE\_RECHERCHE{#URL\_PAGE{xxxx.html}}

Si vous préférez afficher le résultat dans la page qui a applié le formulaire, il faut indiquer l'url de la page appellante entre {} : #FORMULAIRE\_RECHERCHE{#SELF}.

Pour afficher le résultat de la recherche, on utilise simplement le critère {recherche} dans une boucle. Il est possible de les trier en fonction de leur pertinence, on utilise pour cela : {par points}{inverse}.

Exemple :

```
<h3> Recherche d'un article</h3>
#FORMULAIRE_RECHERCHE{#SELF}

<B_search>
<h3>Résultat</h3>
<BOUCLE_search(ARTICLES){recherche}{par points}{inverse}>
<h4><a href="#URL_ARTICLE">#TITRE</a></h4>
</BOUCLE_search>
</B_search>
```

La balise #RECHERCHE permet d'afficher la requête envoyée au moteur de recherche.

Il reste encore à modifier l'apparence du formulaire, vous devez :

1. créer un dossier nommé « formulaires » dans votre dossier « squelettes » ;

2. y copier le fichier « recherche.html » du dossier « squelettes-dis/formulaires » ;
3. modifier le formulaire pour appliquer les bons styles css.

Voici le code minimum du formulaire de recherche :

```
<form action="#ENV{action}" method="get" class="...">>
  [#ENV{action}|form_hidden]
  [<input type="hidden" name="lang" value="#ENV{lang}"]>
  <!--<label
    for="#ENV{_id_champ}"><:info_rechercher_02:></label>-->
  <input type="text" class="..." placeholder="Search"
    name="recherche"
    id="#ENV{_id_champ}" [ value="#ENV{recherche}" ]
    accesskey="4" autocapitalize="off"
    autocorrect="off" onblur="if (this.value == '')"
      {this.value = 'Search';}">
  <button type="submit" class="..." value="..."
    title="<:info_rechercher:>">Submit</button>
</form>
```

### 6.16.2 Formulaire de forum

La balise #FORMULAIRE\_FORUM affiche le formulaire permettant de poster un commentaire. Le formulaire renvoie automatiquement vers la page appellante.

Si la balise est placée dans une boucle article, le commentaire est associé à l'article.

Pour répondre à un commentaire, il faut donc placer la balise dans la boucle affichant le commentaire. Afficher un formulaire sous chaque commentaire ne sera pas lisible. Imaginez un formulaire sous chaque post de la figure ci-dessous.



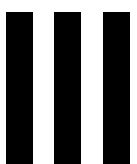
La balise #PARAMETRES\_FORUM va nous permettre de créer un lien permettant d'afficher le formulaire de réponse à un commentaire.

Le lien fabriqué comme ci-dessous envoie vers le squelette nommé « forum.html ».

```
<a href="#URL_PAGE{forum}&#PARAMETRES_FORUM"> Répondre </a>
```

Un exemple de squelette de réponse peut être :

```
<h3>Vous répondez au message suivant :</h3>
<BOUCLE_com(FORUMS){id_forum}>
#TEXTE
</BOUCLE_com>
#FORMULAIRE_FORUM
```



**SGBD**



2<sup>ème</sup> année bachelier en techniques graphiques,  
finalité techniques infographiques

## Modèle relationnel & SQL

Introduction aux réseaux et aux bases de données

F. Gerard

Haute École de la province de Liège,  
catégorie technique

2014-2015

F. Gerard (HEPL)

Intro R&B

2014-2015

1 / 140

## Sommaire général

- ① Chapitre 1 - Modèle relationnel
- ② Chapitre 2 - Comment placer une clé étrangère
- ③ Chapitre 3 - SQL, manipulation de données

F. Gerard (HEPL)

Intro R&B

2014-2015

2 / 140



## Chapitre 1 - Modèle relationnel

### Sommaire

Chapitre 1 - Modèle relationnel

① Introduction

② Modèle relationnel

## Types de SGBD

- A l'heure actuelle, les SGBD les plus utilisés sont de type relationnel.
- Cela signifie qu'ils reposent sur un modèle de conception que l'on nomme modèle relationnel.
- Il existe évidemment d'autre type de SGBD (hiérarchique, réseaux, ...), les modèles utilisés par ces SGBD ne sont plus d'actualité ou ne correspondent pas à nos besoins.

## Sommaire

Chapitre 1 - Modèle relationnel

① Introduction

② Modèle relationnel

## Modèle relationnel

- Le modèle relationnel est une référence formelle à des propositions établies par Edgar Frank Codd en 1970.
- Ce modèle ne fait aucune référence explicite à des notions informatiques (méthode d'accès aux fichiers, organisation des données, ...).
- En bref, ce modèle repose sur des règles mathématiques (algèbre relationnel), il n'y a donc pas d'interprétation ambiguë.

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire.
- Clé étrangère.

### Contraintes

- Intégrité de domaine.
- Intégrité de relation.
- Intégrité de référence.

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

Modèle relationnel    Objets

## Sommaire

Chapitre 1 - Modèle relationnel

- ① Introduction
- ② Modèle relationnel
  - Objets
  - Contraintes
  - Opérateurs

Modèle relationnel    Objets

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

- | <u>Objets</u>  | <u>Contraintes</u>   |
|--|--|
| <ul style="list-style-type: none"><li>• Table (relation).</li><li>• Domaine / attribut.</li><li>• Clé primaire.</li><li>• Clé étrangère.</li></ul> | <ul style="list-style-type: none"><li>• Intégrité de domaine.</li><li>• Intégrité de relation.</li><li>• Intégrité de référence.</li></ul> |

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Table - attribut - domaine

- Une base de données relationnelle, est constituée de une à plusieurs tables.
- Une table est composée de un à plusieurs attributs (colonnes).
- Pour chaque attribut, on définit un ensemble de valeurs autorisées que l'on nomme domaine.

Un exemple de table :

AUTEUR		
NUM_AUTEUR	NOM	PRENOM
1	Leloup	Roger
2	Cauvin	Raoul
3	Franquin	Andre

## Table - attribut - domaine

- Pour l'attribut NUM\_AUTEUR, le domaine pourrait être l'ensemble des entiers positifs.
- Pour l'attribut NOM, le domaine pourrait être l'ensemble des mots qui constituent des noms de familles.
- Pour l'attribut PRENOM, le domaine pourrait être l'ensemble des mots qui constituent des prénoms.

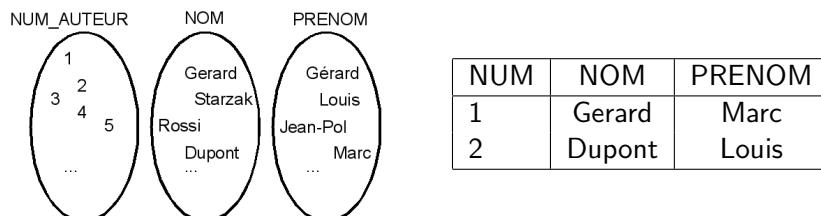
On ne devrait pas voir ceci :

AUTEUR		
NUM_AUTEUR	NOM	PRENOM
a127	Leloup	Roger99
12	C@uvin	Raoul !!

Modèle relationnel    Objets

## Table - attribut - domaine

- Pour chaque attribut, on a donc un domaine de valeurs possibles.
- Chaque ligne d'une table est donc une "association" de valeurs issues des différents domaines.
- En mathématique cela se note ( $v_1, v_2, v_3, \dots$ ) et se nomme un n-uple ou un tuple.
- Lorsque l'on parle d'un tuple, on parle simplement d'une ligne d'une table.



Modèle relationnel    Objets

## Table - attribut - domaine

Une exemple de base de données :

OUVRAGE			
NUM_OUVRAGE	TITRE	ANNEE	RELIURE
1	LE DAVID	1985	LUXE
2	LE TRIO DE L'ÉTRANGE	1980	NORMAL
3	LE CAS LAGAFFE	1980	NORMAL
4	LE GANG DES GAFFEURS	1981	LUXE

AUTEUR			A_ECRIT	
NUM_AUTEUR	NOM	PRENOM	NUM_AUTEUR	NUM_OUVRAGE
1	Leloup	Roger	1	2
2	Cauvin	Raoul	2	1
3	Franquin	Andre	3	3
			3	4

## Table - attribut - domaine

- Pour que la comparaison de deux attributs aie un sens, ils doivent être définis sur le même domaine.
- NUM\_AUTEUR de la table AUTEUR et de la table A\_ECRIT sont comparables.
- NUM\_AUTEUR et NUM\_OUVRAGE de la table A\_ECRIT ne sont pas comparables.
- La possibilité d'effectuer des comparaisons entre deux tables permet simplement de "recouper" les données.

## Table - attribut - domaine

Exemple de "recouplement"

- Leloup Roger a écrit l'ouvrage "2".
- Cauvin Raoul a écrit les ouvrages "3" et "4".
- Franquin Andre a écrit l'ouvrage "1".

AUTEUR			A_ECRIT	
NUM_AUTEUR	NOM	PRENOM	NUM_AUTEUR	NUM_OUVRAGE
1	Leloup	Roger	1	2
2	Cauvin	Raoul	2	1
3	Franquin	Andre	3	3
			3	4

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- **Clé primaire.**
- Clé étrangère.

### Contraintes

- Intégrité de domaine.
- Intégrité de relation.
- Intégrité de référence.

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Clé primaire

- Dans une table, un attribut ou un groupe d'attributs qui permettent d'identifier chaque tuple d'une table se nomme "clé primaire".
- Une table doit toujours avoir une clé primaire.
- De manière simpliste, on dit que : une clé primaire, c'est une ou plusieurs colonnes d'une table qui permettent de faire en sorte que chaque ligne d'une table ne soit pas confondue avec une autre.

## Clé primaire

Exemple de table avec une colonne clé primaire :

NUM	NOM	PRENOM	ADRESSE	...
1	Gerard	Marc	rue pont d'Avroy 26	
2	Dupont	Louis	rue pont d'Avroy 24	
2	Dupont	Louis	rue des écoles 17	

Exemple de table avec 3 colonnes qui forment la clé primaire :

NOM	PRENOM	ADRESSE	...
Gerard	Marc	rue pont d'Avroy 26	
Dupont	Louis	rue pont d'Avroy 24	
Dupont	Louis	rue des écoles 17	

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire
- **Clé étrangère.**

### Contraintes

- Intégrité de domaine.
- Intégrité de relation.
- Intégrité de référence.

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Clé étrangère

Lorsque qu'une table reprend un attribut (ou groupe d'attributs), qui est clé primaire dans une autre table, on nomme cet attribut (ou groupe d'attributs) clé étrangère.

AUTEUR

NUM_AUTEUR	NOM	PRENOM
1	Leloup	Roger
2	Cauvin	Raoul
3	Franquin	Andre

A\_ECRIT

NUM_AUTEUR	NUM_OUVRAGE
1	2
2	1
3	3
3	4

## Clé étrangère

Dans la table A\_ECRIT, les colonnes NUM\_AUTEUR et NUM\_OUVRAGE sont des clés étrangères.

Cela parce que NUM\_AUTEUR et NUM\_OUVRAGE sont clés primaires dans une autre table.

OUVRAGE

NUM_OUVRAGE	TITRE	ANNEE	RELIURE
1	LE DAVID	1985	LUXE
2	LE TRIO DE L'ÉTRANGE	1980	NORMAL
3	LE CAS LAGAFFE	1980	NORMAL
4	LE GANG DES GAFFEURS	1981	LUXE

AUTEUR

NUM_AUTEUR	NOM	PRENOM
1	Leloup	Roger
2	Cauvin	Raoul
3	Franquin	Andre

A\_ECRIT

NUM_AUTEUR	NUM_OUVRAGE
1	2
2	1
3	3
3	4

## Clé étrangère

Exemple d'une clé multi-colonne

Table client :

NOM	PRENOM	ADRESSE	...
Gerard	Marc	rue pont d'Avroy 26	
Dupont	Louis	rue pont d'Avroy 24	
Dupont	Louis	rue des écoles 17	

Table emprunt :

ISBN	NOM	PRENOM	ADRESSE	DATE
12-5897-65	Gerard	Marc	rue pont d'Avroy 26	mai 05
65-8975-68	Dupont	Louis	rue pont d'Avroy 24	juin 05
56-4157-32	Dupont	Louis	rue des écoles 17	aout 08
12-5897-65	Gerard	Marc	rue pont d'Avroy 26	mai 09

## Sommaire

Chapitre 1 - Modèle relationnel

① Introduction

② Modèle relationnel

Objets

Contraintes

Opérateurs

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire.
- Clé étrangère.

### Contraintes

- **Intégrité de domaine.**
- Intégrité de relation.
- Intégrité de référence.

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Contrainte d'intégrité de domaine

- Une contrainte d'intégrité est une règle qui va permettre à la base de données de rester cohérente.
- Un domaine est l'ensemble des valeurs autorisées pour un attribut.
- L'intégrité de domaine permet de vérifier que seules les valeurs appartenant au domaine de l'attribut sont ajoutées dans une table.
- Si l'on veut ajouter la ligne (1,'Lebon127','Hugo') dans la table AUTEUR, le SGBD va refuser.  
'Lebon127' contient des chiffres, cette valeur ne fait donc pas partie du domaine sur lequel la colonne NOM est définie.

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire.
- Clé étrangère.

### Contraintes

- Intégrité de domaine.
- **Intégrité de relation.**
- Intégrité de référence.

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Contrainte d'intégrité de relation

- Les lignes d'une table sont identifiables par la clé primaire.
- L'intégrité de relation permet de vérifier que les valeurs de la clé primaire d'une table sont **uniques et toujours définies**.
- C'est à dire que l'on ne peut pas laisser vide une colonne faisant partie d'une clé primaire.
- Mais également que l'on ne peut pas retrouver la même valeur plusieurs fois dans une clé primaire.

## Contrainte d'intégrité de relation

### Exemple

NOM	PRENOM	ADRESSE	...
Gerard	Marc	rue pont d'Avroy 26	
Dupont	Louis	rue pont d'Avroy 24	
Dupont	Louis	rue des écoles 17	
Gerard	Marc	rue pont d'Avroy 26	
Gerard	Marc	rue pont d'Avroy 26	

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire.
- Clé étrangère.

### Contraintes

- Intégrité de domaine.
- Intégrité de relation.
- **Intégrité de référence.**

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Contrainte d'intégrité de référence

- Une clé étrangère, est une ou plusieurs colonnes qui se réfèrent à une ou plusieurs colonnes clés primaires.
- L'intégrité de référence indique que les valeurs contenues dans les colonnes qui forment la clé étrangère doivent :
  - soit exister dans les colonnes "clé primaire" correspondantes,
  - soit être la valeur inconnue.
- En base de données, une valeur inconnue se note NULL.

### Attention

cette notion n'est pas toujours présente !!!

## Contrainte d'intégrité de référence

### Exemple

Table client :

NOM	PRENOM	ADRESSE	...
Gerard	Marc	rue pont d'Avroy 26	
Dupont	Louis	rue pont d'Avroy 24	
Dupont	Louis	rue des écoles 17	

Table emprunt :

ISBN	NOM	PRENOM	ADRESSE	DATE
12-5897-65	Gerard	Marc	rue pont d'Avroy 26	mai 05
65-8975-68	Dupont	Alfred	rue pont d'Avroy 24	juin 05
56-4157-32	Dupont	Louis	rue des écoles 17	aout 08
12-5897-65				mai 09

Modèle relationnel    Opérateurs

## Sommaire

Chapitre 1 - Modèle relationnel

① Introduction

② Modèle relationnel

    Objets  
    Contraintes  
    Opérateurs

Modèle relationnel    Opérateurs

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire.
- Clé étrangère.

### Contraintes

- Intégrité de domaine.
- Intégrité de relation.
- Intégrité de référence.

### Opérateurs

- **Opérateurs sémantiques (liés aux domaines).**
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Les opérateurs sémantiques

- SQL est un langage qui permet de manipuler les SGBD.
- Les opérateurs sémantiques sont la partie de SQL qui permet de créer les domaines.

### Attention

Cette notion n'existe pas encore dans les SGBD.

Toutefois, lorsque l'on crée une table on donne un type à chaque colonne. Ce type (entier, du texte, nombre réel, ...) permet, en partie du moins, de remplacer les domaines.

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire.
- Clé étrangère.

### Contraintes

- Intégrité de domaine.
- Intégrité de relation.
- Intégrité de référence.

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- **Opérateurs ensemblistes (produit cartésien, union, ...).**
- Opérateurs relationnels (sélection, projection, jointure, ...).

## Les opérateurs ensemblistes

- Il s'agit des opérations d'union, de différence et de produit cartésien que vous avez rencontré pendant vos études primaires.

### Produit cartésien

Le Produit cartésien de deux tables est une table ayant pour attributs tous les attributs des deux tables.

Les tuples qui constituent la table résultat sont formés de toutes les concaténations possibles d'un tuple de la table1 à un tuple de la table2.

## Les opérateurs ensemblistes

exemple produit cartésien

AUTEUR		
NUM_AUTEUR	NOM	PRENOM
1	Leloup	Roger
2	Cauvin	Raoul
3	Franquin	Andre

A_ECRIT	
NUM_AUTEUR	NUM_OUVRAGE
1	2
2	1
3	3
3	4

NUM_AUTEUR	NOM	PRENOM	NUM_AUTEUR	NUM_OUVRAGE
1	Leloup	Roger	1	2
1	Leloup	Roger	2	1
1	Leloup	Roger	3	3
1	Leloup	Roger	3	4
2	Cauvin	Raoul	1	2
2	Cauvin	Raoul	2	1
2	Cauvin	Raoul	3	3
2	Cauvin	Raoul	3	4
3	Franquin	Andre	1	2
3	Franquin	Andre	2	1
3	Franquin	Andre	3	3
3	Franquin	Andre	3	4

## Les concepts du modèle relationnel

Le modèle relationnel repose sur 3 piliers.

### Objets

- Table (relation).
- Domaine / attribut.
- Clé primaire.
- Clé étrangère.

### Contraintes

- Intégrité de domaine.
- Intégrité de relation.
- Intégrité de référence.

### Opérateurs

- Opérateurs sémantiques (liés aux domaines).
- Opérateurs ensemblistes (produit cartésien, union, intersection, ...).
- **Opérateurs relationnels (sélection, projection, jointure, ...).**

## Les opérateurs relationnels

- L'opérateur de *projection* permet de "cacher" certaines colonnes d'une table.
- L'opérateur de *sélection* permet de "cacher" certaines lignes d'une table, selon un ou plusieurs critères.

Un critère de sélection c'est par exemple :

- NUM\_OUVRAGE > 3 ;
- (NUM\_OUVRAGE < 2 et ANNEE = 1989) ou ANNEE <= 1990.

OUVRAGE

NUM_OUVRAGE	TITRE	ANNEE	RELIURE
1	LE DAVID	1985	LUXE
2	LE TRIO DE L'ÉTRANGE	1980	NORMAL
3	LE CAS LAGAFFE	1980	NORMAL
4	LE GANG DES GAFFEURS	1981	LUXE

## Les opérateurs relationnels

- L'opérateur de *jointure* permet de simplifier la combinaison du produit cartésien et de la sélection.
- Cet opérateur va donc permettre d'associer 2 ou plusieurs tables et de "cacher" les lignes qui ne nous conviennent pas.
- Et également de cacher la clé étrangère.

## Les opérateurs additionnels

Produit cartésien

NUM_AUTEUR	NOM	PRENOM	NUM_AUTEUR	NUM_OUVRAGE
1	Leloup	Roger	1	2
1	Leloup	Roger	2	1
1	Leloup	Roger	3	3
1	Leloup	Roger	3	4
2	Cauvin	Raoul	1	2
2	Cauvin	Raoul	2	1
2	Cauvin	Raoul	3	3
2	Cauvin	Raoul	3	4
3	Franquin	Andre	1	2
3	Franquin	Andre	2	1
3	Franquin	Andre	3	3
3	Franquin	Andre	3	4

Jointure

NUM_AUTEUR	NOM	PRENOM	NUM_OUVRAGE
1	Leloup	Roger	2
2	Cauvin	Raoul	2
2	Cauvin	Raoul	1
3	Franquin	Andre	3
3	Franquin	Andre	4

## Chapitre 2 - Comment placer une clé étrangère

### Comment placer une clé étrangère

- Nous voulons créer une BD qui permet d'associer des personnes à leurs voitures.
- Où faut-il placer une colonne supplémentaire pour créer la clé étrangère ?
- Plusieurs cas sont à considérer.

Personne

Num_registro	Nom	...
090105-253-16	Dubus	...
762409-007-01	Gaspard	...
662567-951-65	Dumont	...
900502-542-86	Dupond	...

Voiture

Num_chassi	Marque	...
azz46658ra	Ford	...
a4srt45556	Renault	...
er54fzerzr	Citroën	...

0,1 - 0,N

## Sommaire

Chapitre 2 - Comment placer une clé étrangère

③ 0,1 - 0,N

④ 1,1 - 0,N

⑤ 0,1 - 0,1

⑥ 1,1 - 0,1

⑦ 0,N - 0,N

⑧ En résumé

0,1 - 0,N

## Association : 0,1 - 0,N

Clé étrangère à 1 colonne

L'association 0,1 - 0,N peut se traduire comme ceci :

- 1 personne possède soit 0, soit 1 voiture ;
- 1 voiture est possédée soit par 0, soit par 1, soit plusieurs personnes.

Personne

Num_registro	Nom	...
090105-253-16	Dubus	...
762409-007-01	Gaspard	...
662567-951-65	Dumont	...
900502-542-86	Dupond	...

Voiture

Num_chassi	Marque	...	Num_registro
azz46658ra	Ford	...	090105-253-16
a4srt45556	Renault	...	090105-253-16
er54fzerzr	Citroën	...	NULL

- 1 Personne → 0 voiture : ok
- 1 Personne → 1 voiture : ok
- 1 Personne → N voitures : ko
- 1 voiture → 0 personne : ok
- 1 voiture → 1 personne : ok
- 1 voiture → N personne : ko

0,1 - 0,N

## Association : 0,1 - 0,N

Clé étrangère à 1 colonne

Plaçons la clé étrangère dans la table personne.

Personne

Num_registro	Nom	...	Num_chassi
090105-253-16	Dubus	...	azz46658ra
762409-007-01	Gaspard	...	a4srt45556
662567-951-65	Dumont	...	a4srt45556
900502-542-86	Dupond	...	NULL

Voiture

Num_chassi	Marque	...
azz46658ra	Ford	...
a4srt45556	Renault	...
er54fzerzr	Citroën	...

- 1 Personne → 0 voiture : ok
- 1 Personne → 1 voiture : ok
- 1 Personne → N voitures : ko

- 1 voiture → 0 personne : ok
- 1 voiture → 1 personne : ok
- 1 voiture → N personnes : ok

0,1 - 0,N

## Association : 0,1 - 0,N

Clé étrangère à 1 colonne

Personne	
Num_registro	
Nom	
Prenom	
Date_Naissance	
...	
Num_chassi	

Voiture	
Num_chassi	
Marque	
Modèle	
Couleur	
...	

## Association : 0,1 - 0,N

Clé étrangère multi-colonnes

- Nous voulons créer une BD qui permet d'associer 1 personne à son logement (0, 1 ou plusieurs).

Personne

Num_registro	...	Rue	...	code_postal
090105-253-16	...	des écoles	...	4100
762409-007-01	...	du pot d'or	...	4000
782815-896-14	...	du pot d'or	...	4000
900502-542-86	...	NULL	...	NULL

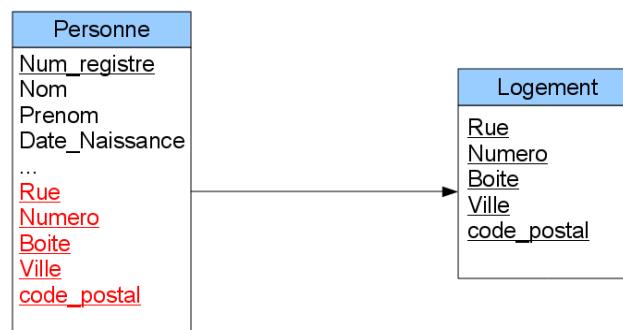
Logement

Rue	...
du pont	...
du pot d'or	...
des écoles	...

## Association : 0,1 - 0,N

Clé étrangère multi-colonnes

Ici, la clé primaire de logement est multi-colonnes, la clé étrangère comporte donc les mêmes colonnes.



1,1 - 0,N

## Sommaire

Chapitre 2 - Comment placer une clé étrangère

③ 0,1 - 0,N

④ 1,1 - 0,N

⑤ 0,1 - 0,1

⑥ 1,1 - 0,1

⑦ 0,N - 0,N

⑧ En résumé

1,1 - 0,N

## Association : 1,1 - 0,N

Clé étrangère à 1 colonne

Une personne doit toujours avoir une voiture.

Personne

Num_registro	Nom	...	Num_chassi
090105-253-16	Dubus	...	azz46658ra
762409-007-01	Gaspard	...	a4srt45556
662567-951-65	Dumont	...	a4srt45556
900502-542-86	Dupond	...	NULL

Voiture

Num_chassi	Marque	...
azz46658ra	Ford	...
a4srt45556	Renault	...
er54fzerzr	Citroën	...

- 1 Personne → 0 voiture : ko

- 1 Personne → 1 voiture : ok

- 1 Personne → N voitures : ko

- 1 voiture → 0 personne : ok

- 1 voiture → 1 personne : ok

- 1 voiture → N personnes : ok

## Association : 1,1 - 0,N

Clé étrangère à 1 colonne

Il faut interdire la valeur NULL dans la colonne clé étrangère.



## Association : 1,1 - 0,N

Clé étrangère multi-colonnes

- Une personne doit toujours avoir un logement,
- la valeur NULL doit donc être interdite dans les colonnes qui forment la clé étrangère.

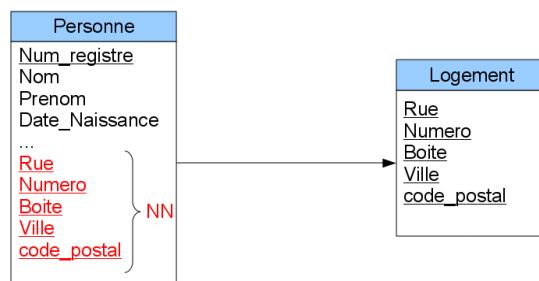
Personne				
Num_registro	...	Rue	...	code_postal
090105-253-16	...	des écoles	...	4100
762409-007-01	...	du pot d'or	...	4000
662567-951-65	...	du pot d'or	...	4000
900502-542-86	...	NULL	...	NULL

Logement	
Rue	...
du pont	...
du pot d'or	...
des écoles	...

1,1 - 0,N

## Association : 1,1 - 0,N

Clé étrangère multi-colonnes



0,1 - 0,1

## Sommaire

Chapitre 2 - Comment placer une clé étrangère

③ 0,1 - 0,N

④ 1,1 - 0,N

⑤ 0,1 - 0,1

⑥ 1,1 - 0,1

⑦ 0,N - 0,N

⑧ En résumé

0,1 - 0,1

## Association : 0,1 - 0,1

Clé étrangère à 1 colonne

- Une personne possède soit 0, soit 1 voiture.
- Une voiture est possédée soit par 0, soit par 1 personne.

Personne

Num_registro	Nom	...	Num_chassi
090105-253-16	Dubus	...	azz46658ra
762409-007-01	Gaspard	...	a4srt45556
662567-951-65	Dumont	...	a4srt45556
900502-542-86	Dupond	...	NULL

Voiture

Num_chassi	Marque	...
azz46658ra	Ford	...
a4srt45556	Renault	...
er54fzerzr	Citroën	...

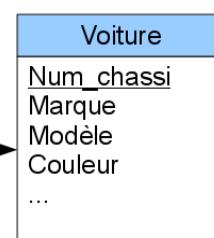
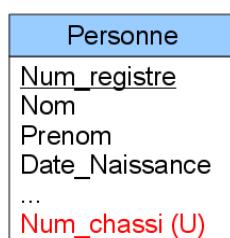
- 1 Personne → 0 voiture : ok
- 1 Personne → 1 voiture : ok
- 1 Personne → N voitures : ko
- 1 voiture → 0 personne : ok
- 1 voiture → 1 personne : ok
- 1 voiture → N personnes : ok

0,1 - 0,1

## Association : 0,1 - 0,1

Clé étrangère à 1 colonne

La clé étrangère ne peut pas contenir plusieurs fois la même valeur, on dit que la colonne est marquée UNIQUE.



### Attention

Uniquement dans le cas des associations 0,1 - 0,1.

La table dans laquelle on place la clé étrangère n'a pas d'incidence.

0,1 - 0,1

## Association : 0,1 - 0,1

Clé étrangère multi-colonnes

Dans le cas d'une clé étrangère multi-colonnes, c'est l'ensemble des colonnes qui forment la clé qui doivent être marquées comme UNIQUE.

Personne

Num_registro	...	Rue	...	code_postal
090105-253-16	...	des écoles	...	4100
762409-007-01	...	du pot d'or	...	4000
662567-951-65	...	du pot d'or	...	4000
900502-542-86	...	NULL	...	NULL

Logement

Rue	...
du pont	...
du pot d'or	...
des écoles	...

0,1 - 0,1

## Association : 0,1 - 0,1

Clé étrangère multi-colonnes

Personne

Num_registro
Nom
Prenom
Date_Naissance
...
<b>Rue</b>
<b>Numero</b>
<b>Boite</b>
<b>Ville</b>
<b>code_postal</b>

Voiture

Rue
Numero
Boite
Ville
code_postal

1,1 - 0,1

## Sommaire

Chapitre 2 - Comment placer une clé étrangère

③ 0,1 - 0,N

④ 1,1 - 0,N

⑤ 0,1 - 0,1

⑥ 1,1 - 0,1

⑦ 0,N - 0,N

⑧ En résumé

1,1 - 0,1

## Association : 1,1 - 0,1

Clé étrangère à 1 colonne

- Une personne doit toujours avoir une voiture.
- Une voiture est possédée soit par 0, soit par 1 personne.

Personne

Num_registro	Nom	...	Num_chassi
090105-253-16	Dubus	...	azz46658ra
762409-007-01	Gaspard	...	a4srt45556
662567-951-65	Dumont	...	a4srt45556
900502-542-86	Dupond	...	NULL

Voiture

Num_chassi	Marque	...
azz46658ra	Ford	...
a4srt45556	Renault	...
er54fzerzr	Citroën	...

- ~~1 Personne → 0 voiture~~ : ok
- ~~1 Personne → 1 voiture~~ : ok
- ~~1 Personne → N voitures~~ : ko
- 1 voiture → 0 personne : ok
- 1 voiture → 1 personne : ok
- 1 voiture → N personnes : ok

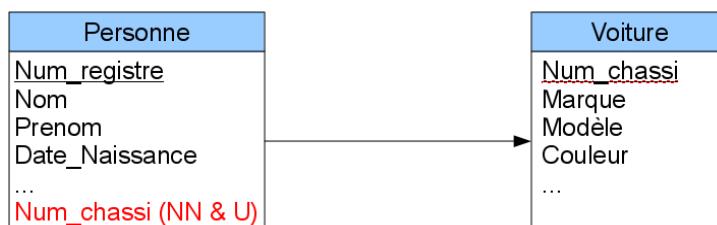
1,1 - 0,1

## Association : 1,1 - 0,1

Clé étrangère simple

Il faut interdire la valeur Null, mais également interdire les doublons dans la clé étrangère.

La colonne clé étrangère doit donc être NON NULL et UNIQUE.



1,1 - 0,1

## Association : 1,1 - 0,1

Clé étrangère multi-colonnes

Dans le cas d'une clé multi-colonnes, c'est l'ensemble des colonnes qui forment la clé qui doivent être marquées comme UNIQUE et NON NULL.

Personne

Num_register	...	Rue	...	code_postal
090105-253-16	...	des écoles	...	4100
762409-007-01	...	du pot d'or	...	4000
662567-951-65	...	du-pot-d'or	...	4000
900502-542-86	...	NULL	...	NULL

Logement

Rue	...
du pont	...
du pot d'or	...
des écoles	...

0,N - 0,N

## Sommaire

Chapitre 2 - Comment placer une clé étrangère

③ 0,1 - 0,N

④ 1,1 - 0,N

⑤ 0,1 - 0,1

⑥ 1,1 - 0,1

⑦ 0,N - 0,N

⑧ En résumé

0,N - 0,N

## Association : 0,N - 0,N

Clé étrangère à 1 colonne

- Une personne possède soit 0, soit 1, soit plusieurs voitures.
- Une voiture est possédée soit par 0, soit par 1, soit plusieurs personnes.

Personne	
Num_registro	...
090105-253-16	...
762409-007-01	...
900502-542-86	...

perso_voiture	
Num_registro	Num_chassi
090105-253-16	a4srt45556
090105-253-16	er54fzerzr
900502-542-86	a4srt45556
900502-542-86	er54fzerzr

Voiture	
Num_chassi	...
azz46658ra	...
a4srt45556	...
er54fzerzr	...

### Attention

Ici, on n'a pas d'autre choix que d'ajouter une nouvelle table.  
On la nomme généralement table d'association.

0,N - 0,N

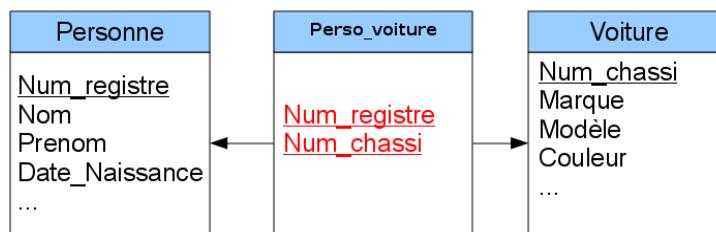
## Association : 0,N - 0,N

Clé étrangère à 1 colonne

### Attention

Chaque colonne de la table perso\_voiture est une clé étrangère vers sa table respective.

C'est l'ensemble des deux colonnes qui forment la clé primaire de la table d'association.



0,N - 0,N

## Association : 0,N - 0,N

Clé étrangère multi-colonnes

- Une personne possède soit 0, soit 1, soit plusieurs logement.
- Une logement est possédée soit par 0, soit par 1, soit plusieurs personnes.

Personne	
Num_register	...
090105-253-16	...
762409-007-01	...
900502-542-86	...

Louer		
Num_register	Rue	...
090105-253-16	du pont	...
090105-253-16	des écoles	...
900502-542-86	des écoles	...

Logement	
Rue	...
du pont	...
du pot d'or	...
des écoles	...



## Sommaire

Chapitre 2 - Comment placer une clé étrangère

③ 0,1 - 0,N

④ 1,1 - 0,N

⑤ 0,1 - 0,1

⑥ 1,1 - 0,1

⑦ 0,N - 0,N

⑧ En résumé

En résumé

## En résumé

Où placer la clé étrangère :

- 0,1 - 0,N : du côté 0,1.
- 1,1 - 0,N : du côté 1,1 avec la clé étrangère **NOT NULL**.
- 0,1 - 0,1 : où l'on veut avec la clé étrangère **UNIQUE**.
- 1,1 - 0,1 : du côté 1,1 avec la clé étrangère **NOT NULL & UNIQUE**.
- 0,N - 0,N : il faut créer une nouvelle table qui reprend les clés primaires des deux tables.

Chaque colonne étant une clé étrangère vers la table qui lui correspond.

L'ensemble de ces colonnes forment la clé primaire de la table d'association.

## Chapitre 3 - SQL, manipulation de données

## Sommaire

Chapitre 3 - SQL, manipulation de données

9 SQL

16 Renommage - Alias

10 Projections

17 ORDER BY

11 Sélections

18 GROUP BY

12 Fonctions de calcul

19 Ajout de données

13 Opérations arithmétiques

20 Mise à jour de données

14 Dates et Heures

21 Supprimer des données

Jointures

F. Gerard (HEPL)

Intro R&B

2014-2015

73 / 140

## SQL : Structured Query Language

SQL est un langage permettant de créer, manipuler et de contrôler une base de données relationnelle.

Ce n'est pas un langage de programmation proprement dit !

- Il peut être inséré dans des langages de programmation comme le C, le PHP, ...
- Il peut encore être utilisé en mode interactif, c'est à dire que l'on se connecte directement à la base de données et que l'on communique directement en SQL avec elle.

## SQL : Structured Query Language

SQL est divisé en trois grandes parties :

- DML : Data Manipulation language.  
C'est la partie qui est utilisée couramment.
- DDL : Data Definition language.  
Cette partie permet de créer les tables.
- DCL : Data Control language.  
Cette partie permet d'administrer le SGBD.

## SQL : Structured Query Language

SQL est insensible à la case, que l'on écrive :

- SELECT \* from table ;
- Select \* From Table ;
- selECt \* from taBLe ;

cela revient au même.

## Convention utilisée dans les notes

Par convention :

- les mots clés seront toujours en lettres majuscules ;
- les parties optionnelles seront entre [ ] ;
- le symbole | signifie : ou.

## Attention

Attention Attention Attention Attention Attention Attention

Toutes les requêtes SQL sans exception doivent se terminer par un ;

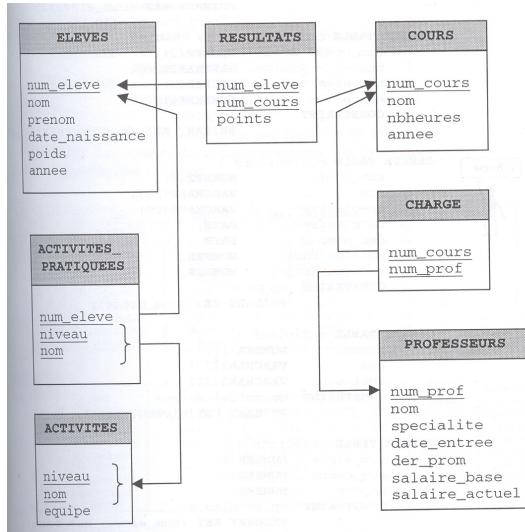
Voici comment écrire correctement une requête en PHP :

```
$sql = "SELECT * FROM eleves;" ;
```

Le premier ; fait partie de la requête SQL, le second est le point virgule qui doit terminer toutes lignes de code en PHP.

SQL

## Schéma de la BD utilisée pour les notes



SQL

## Schéma de la BD utilisée pour les notes

Table *eleves*

num_eleve	nom	prenom	date_naissance	poids	annee
1	Brisefer	Benoit	1978-12-10	35	1
2	Genial	Olivier	1978-04-10	42	1
3	Jourdan	Gil	1974-06-28	72	2
4	Spring	Jerry	1974-02-16	78	2
5	Tsuno	Yoko	1977-10-29	45	1
6	Lebut	Marc	1974-04-29	75	2
7	Lagaffe	Gaston	1975-04-08	61	1
8	Dubois	Robin	1976-04-20	60	2
9	Walthery	Natache	1977-09-07	59	1
10	Danny	Buck	1973-02-15	82	2

SQL

## Schéma de la BD utilisée pour les notes

Table *professeurs*

num_prof	nom	specialite	date_entree	der_prom	...
1	Bottle	poésie	1970-10-01	1988-10-01	...
2	Bolenov	réseaux	1968-11-15	1998-10-01	...
3	Tonilaclasse	poo	1979-10-01	1989-01-10	...
4	Pastecnov	sql	1975-10-01	NULL	...
5	Selector	sql	1982-10-15	NULL	...
6	Vilplusplus	poo	1990-04-25	1994-06-05	...
7	Francesca	NULL	1975-10-01	1998-01-11	...
8	Pucette	sql	1988-12-06	1996-02-29	...

Projections

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑯ Renommage - Alias

⑩ Projections

⑭ ORDER BY

⑪ Sélections

⑮ GROUP BY

⑫ Fonctions de calcul

⑯ Ajout de données

⑬ Opérations arithmétiques

⑰ Mise à jour de données

⑭ Dates et Heures

㉑ Supprimer des données

㉒ Jointures

## Afficher le contenu d'une table

```
SELECT * FROM nom_table;
```

- \* : pour afficher toutes les colonnes
- nom\_table : est le nom de la table à afficher
- le résultat est une table

### Exemples

- SELECT \* FROM eleves ;
- SELECT \* FROM professeurs ;

## Afficher une ou plusieurs colonnes d'une table

```
SELECT liste_colonnes FROM nom_table;
```

- liste\_colonnes : nom\_colonne1 [, nom\_colonne2, ...]
- ceci correspond à l'opérateur de projection
- le résultat est une colonne (une liste de valeurs) si on ne choisit qu'une seule colonne, sinon le résultat est une table.

### Exemples

- SELECT nom FROM eleves ;
- SELECT nom, poids FROM eleves ;

## Projections

## Éliminer les lignes doublons

```
SELECT DISTINCT clause_de_projection FROM nom_table;
```

- DISTINCT : fait en sorte de ne pas afficher les lignes doublons dans le résultat
- clause\_de\_projection : \* | nom\_colonne1 [, nom\_colonne2, ...]

### Comparer les résultats de

- SELECT specialite FROM professeurs ;
- SELECT DISTINCT specialite FROM professeurs ;

## Sélections

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑯ Renommage - Alias

⑩ Projections

⑰ ORDER BY

⑪ Sélections

⑲ GROUP BY

⑫ Fonctions de calcul

⑯ Ajout de données

⑬ Opérations arithmétiques

⑳ Mise à jour de données

⑭ Dates et Heures

㉑ Supprimer des données

⑮ Jointures

## Sélections

## Enlever des lignes qui ne conviennent pas

```
SELECT clause_de_projection FROM nom_table WHERE condition ;
```

Avec condition qui vaut :

- condition\_base
- | condition\_between
- | condition\_in
- | condition\_like
- | condition\_null
- | condition\_AND
- | condition\_OR

## Sélections Condition de base

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

Condition de base

BETWEEN

IN

LIKE

NULL

AND

OR

Conditions multiples

## condition\_base

Les conditions de base sont :

- colonne = constante
- | colonne <> constante
- | colonne < constante
- | colonne <= constante
- | colonne > constante
- | colonne >= constante

### Exemples

- SELECT nom, poids FROM eleves WHERE poids <= 45 ;
- SELECT nom FROM professeurs WHERE specialite = 'poo' ;

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

Condition de base

BETWEEN

IN

LIKE

NULL

AND

OR

Conditions multiples

Sélections    BETWEEN

## condition\_between

La condition between s'écrit :

- colonne BETWEEN constante AND constante

### Exemple

```
SELECT nom, poids FROM eleves WHERE poids BETWEEN 60 AND 80;
```

Sélections    IN

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

Condition de base

BETWEEN

IN

LIKE

NULL

AND

OR

Conditions multiples

## condition\_in

La condition IN s'écrit :

- colonne [NOT] IN (liste\_de\_valeurs)

### Exemples

- `SELECT * FROM professeurs WHERE specialite IN ('poésie','sql');`
- `SELECT * FROM professeurs WHERE specialite NOT IN ('poésie','sql');`

## condition\_in

La liste de valeurs peut provenir d'un SELECT.

### Exemple

```
SELECT * FROM eleves WHERE num_eleve IN  
(SELECT num_eleve FROM activites_pratiquees WHERE niveau = 1 AND  
nom = 'surf');
```

- Attention, la colonne résultat du SELECT doit être cohérente par rapport à la colonne indiquée à gauche du IN.
- Lorsque l'on place une requête SQL à l'intérieur d'une autre, on parle de requêtes imbriquées.

## Sommaire

Chapitre 3 - SQL, manipulation de données

9 SQL

10 Projections

11 Sélections

- Condition de base
- BETWEEN
- IN
- LIKE
- NULL
- AND
- OR
- Conditions multiples

## condition\_like

La condition LIKE s'écrit :

- colonne [NOT] LIKE 'modèle à rechercher' [ESCAPE escape\_car].

Le modèle peut contenir des caractères quelconques plus deux caractères spéciaux :

- \_ remplace un seul caractère ;
- % remplace un nombre quelconque (éventuellement 0) de caractères.

### Exemples

- SELECT nom FROM eleves WHERE nom LIKE 'Ber%'
- SELECT nom FROM eleves WHERE nom LIKE '%d'
- SELECT nom FROM eleves WHERE nom LIKE '%an%'
- SELECT nom FROM eleves WHERE nom LIKE '%v\_n%'
- SELECT nom FROM eleves WHERE nom LIKE '%d\_'

Sélections    LIKE

## condition\_like

Si l'on veut que le modèle recherché soit par exemple un texte qui se termine par %, il faut alors utiliser la clause ESCAPE.

### Exemple

```
SELECT nom FROM eleves WHERE nom LIKE '%@%' ESCAPE '@'
```

Si l'on veut que le modèle recherché contienne %, n'importe où :

### Exemple

```
SELECT nom FROM eleves WHERE nom LIKE '%@%%' ESCAPE '@'
```

Sélections    NULL

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

Condition de base

BETWEEN

IN

LIKE

NULL

AND

OR

Conditions multiples

Sélections    NULL

## condition\_null

La condition NULL s'écrit :

- colonne IS [NOT] NULL

### Exemples

- SELECT \* FROM professeurs WHERE specialite IS NULL ;
- SELECT \* FROM professeurs WHERE specialite IS NOT NULL ;

Sélections    AND

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

Condition de base

BETWEEN

IN

LIKE

NULL

AND

OR

Conditions multiples

Sélections AND

## condition\_AND

La condition AND s'écrit :

- condition\_1 AND conditon\_2
- Ceci permet de préciser plusieurs conditions.
- Pour que les lignes soient reprises, les deux conditions doivent être vérifiées.

### Exemple

```
SELECT nom FROM eleves WHERE poids < 45 AND annee = 1;
```

Sélections OR

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

Condition de base

BETWEEN

IN

LIKE

NULL

AND

OR

Conditions multiples

## condition\_OR

La condition OR s'écrit :

- condition\_1 OR condition\_2
- Ceci permet également de préciser plusieurs conditions.
- Les lignes du résultat sont celles qui satisfont au moins à une des deux conditions.

### Exemple

```
SELECT nom FROM eleves WHERE poids < 45 OR annee = 2 ;
```

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

Condition de base  
BETWEEN  
IN  
LIKE  
NULL  
AND  
OR

Conditions multiples

Sélections    Conditions multiples

## mélange de conditions

### Exemple

```
SELECT nom FROM élèves  
WHERE (poids BETWEEN 45 AND 60 AND année = 1)  
OR année = 2;
```

Fonctions de calcul

## Sommaire

Chapitre 3 - SQL, manipulation de données

9 SQL

16 Renommage - Alias

10 Projections

17 ORDER BY

11 Sélections

18 GROUP BY

12 Fonctions de calcul

19 Ajout de données

13 Opérations arithmétiques

20 Mise à jour de données

14 Dates et Heures

21 Supprimer des données

Jointures

## Fonctions de calcul

- count(\*)
- count([DISTINCT] nom\_colonne)
- AVG([DISTINCT] nom\_colonne)
- MAX([DISTINCT] nom\_colonne)
- MIN([DISTINCT] nom\_colonne)
- SUM([DISTINCT] nom\_colonne)

## Exemple

```
SELECT AVG(poids) FROM eleves ...
```

## A ne pas faire

```
SELECT AVG( SUM(poids)) FROM eleves ...
```

## Fonctions de calcul

## Fonctions de calcul

- count(\*) prend en compte les NULL ;
- les autres fonctions non.

- Si le résultat de la requête est vide, count donnera 0 ;
- les autres fonctions donneront null.

## Fonctions de calcul

**Fonctions de calcul - exemples**

Obtenir la somme des points de l'élève numéro 3

```
SELECT SUM(points) FROM resultats WHERE num_eleve = 3;
```

Obtenir le poids moyen des élèves de 1ier année

```
SELECT AVG(poids) FROM eleves WHERE annee = 1;
```

Obtenir la plus petite et la plus grande côte de l'élève numéro 1

```
SELECT MIN(points), MAX(points) FROM eleves WHERE num_eleve = 1;
```

Obtenir le nombre d'élèves inscrit en deuxième année

```
SELECT count(*) FROM eleves WHERE annee = 2;
```

## Opérations arithmétiques

**Sommaire**

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑯ Renommage - Alias

⑩ Projections

⑭ ORDER BY

⑪ Sélections

⑮ GROUP BY

⑫ Fonctions de calcul

⑯ Ajout de données

⑬ Opérations arithmétiques

⑰ Mise à jour de données

⑭ Dates et Heures

㉑ Supprimer des données

## Opérations arithmétiques

## Faire des calculs - exemples

Obtenir le nom et salaire mensuel actuel des profs

```
SELECT nom, salaire_actuel / 12 FROM professeurs;
```

Obtenir l'augmentation du salaire mensuel des profs

```
SELECT nom, (salaire_actuel-salaire_base) / 12 FROM professeurs;
```

Obtenir le nom des profs dont l'augmentation du salaire de base dépasse 25%

```
SELECT nom FROM professeurs WHERE (salaire_actuel - salaire_base) / salaire_base > 0.25 ;
```

Obtenir l'augmentation moyenne mensuelle

```
SELECT AVG((salaire_actuel - salaire_base) / 12) FROM professeurs;
```

## Dates et Heures

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑯ Renommage - Alias

⑩ Projections

⑯ ORDER BY

⑪ Sélections

⑯ GROUP BY

⑫ Fonctions de calcul

⑯ Ajout de données

⑬ Opérations arithmétiques

⑯ Mise à jour de données

⑭ Dates et Heures

⑯ Supprimer des données

## Dates et Heures

### Obtenir la date et/ou l'heure du système

- `SELECT CURRENT_DATE;`
- `SELECT CURRENT_TIME;`
- `SELECT CURRENT_TIMESTAMP;`
- `SELECT EXTRACT(champ FROM source);`

champ peut valoir :

- YEAR,
- MONTH,
- DAY,
- HOUR,
- MINUTE,
- SECOND

## Dates et Heures

### Exemples

- `SELECT EXTRACT(YEAR FROM CURRENT_DATE);`
- `SELECT EXTRACT(DAY FROM der_prom) FROM professeurs;`

## Jointures

**Sommaire**

Chapitre 3 - SQL, manipulation de données

[9 SQL](#)[16 Renommage - Alias](#)[10 Projections](#)[17 ORDER BY](#)[11 Sélections](#)[18 GROUP BY](#)[12 Fonctions de calcul](#)[19 Ajout de données](#)[13 Opérations arithmétiques](#)[20 Mise à jour de données](#)[14 Dates et Heures](#)[21 Supprimer des données](#)**15 Jointures**

F. Gerard (HEPL)

Intro R&amp;B

2014-2015

115 / 140

## Jointures

## Produit cartésien

**Sommaire**

Chapitre 3 - SQL, manipulation de données

[9 SQL](#)[10 Projections](#)[11 Sélections](#)[12 Fonctions de calcul](#)[13 Opérations arithmétiques](#)[14 Dates et Heures](#)**15 Jointures**

F. Gerard (HEPL)

Intro R&amp;B

2014-2015

116 / 140

Jointures   Produit cartésien

## Produit cartésien

### Produit cartésien

```
SELECT * from professeurs, charge;
```

### Produit cartésien - autre notation

```
SELECT * from professeurs CROSS JOIN charge;
```

Jointures   Jointures

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

⑫ Fonctions de calcul

⑬ Opérations arithmétiques

⑭ Dates et Heures

⑮ Jointures

## Jointure

### Jointure par le produit cartésien

```
SELECT * from professeurs, charge WHERE professeurs.num_prof =
charge.num_prof ;
```

### Jointure

```
SELECT * from professeurs JOIN charge USING (num_prof) ;
```

### Jointure - si la clé primaire et la clé étrangère n'ont pas le même nom

```
SELECT * from professeurs JOIN charge ON professeurs.num_prof =
charge.num_prof ;
```

## Jointure sur clés multi-colonnes

### Jointure

```
SELECT * from activites JOIN activites_pratiques USING (niveau, nom) ;
```

### Jointure - autre notation

```
SELECT * from activites JOIN activites_pratiques
ON activites.niveau = activites_pratiques.niveau
AND activites.nom = activites_pratiques.nom ;
```

### Jointure par produit cartésien

```
SELECT * from activites, activites_pratiques
WHERE activites.niveau = activites_pratiques.niveau
AND activites.nom = activites_pratiques.nom ;
```

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑩ Projections

⑪ Sélections

⑫ Fonctions de calcul

⑬ Opérations arithmétiques

⑭ Dates et Heures

⑮ Jointures

F. Gerard (HEPL)

Intro R&B

2014-2015

121 / 140

## Jointure de 3 tables

### Jointure

```
SELECT * FROM cours JOIN charge USING (num_cours)
JOIN professeurs USING (num_prof);
```

### Jointure - autre notation

```
SELECT * FROM cours JOIN charge
ON cours.num_cours = charge.num_cours
JOIN professeurs USING (num_prof);
```

## Jointure de 3 tables

### Jointure - autre notation

```
SELECT * FROM cours JOIN charge  
ON cours.num_cours = charge.num_cours  
JOIN professeurs  
ON charge.num_prof=professeurs.num_prof;
```

### Jointure par produit cartésien

```
SELECT * FROM cours, professeurs, charge  
WHERE cours.num_cours = charge.num_cours  
AND charge.num_prof=professeurs.num_prof;
```

## Jointure externe

### Jointure externe

```
SELECT * FROM table1 LEFT OUTER JOIN table2  
USING(col);  
SELECT * FROM table1 RIGHT OUTER JOIN table2  
USING(col);
```

Renommage - Alias

## Sommaire

Chapitre 3 - SQL, manipulation de données

<p><a href="#">9 SQL</a></p> <p><a href="#">10 Projections</a></p> <p><a href="#">11 Sélections</a></p> <p><a href="#">12 Fonctions de calcul</a></p> <p><a href="#">13 Opérations arithmétiques</a></p> <p><a href="#">14 Dates et Heures</a></p>	<p><a href="#">16 Renommage - Alias</a></p> <p><a href="#">17 ORDER BY</a></p> <p><a href="#">18 GROUP BY</a></p> <p><a href="#">19 Ajout de données</a></p> <p><a href="#">20 Mise à jour de données</a></p> <p><a href="#">21 Supprimer des données</a></p>
--	---

[!\[\]\(d916268fdc95a334d01411931d2bd871\_img.jpg\) Jointures](#)

F. Gerard (HEPL)      Intro R&B      2014-2015      125 / 140

Renommage - Alias

### renommage de colonne

- `SELECT count(*) AS nbr FROM eleves;`
- `SELECT salaire_base AS sal FROM professeurs;`
- `SELECT * FROM activites_pratiquees AS ap WHERE ap.niveau = 2;`

[F. Gerard \(HEPL\)](#)      [Intro R&B](#)      [2014-2015](#)      [126 / 140](#)

ORDER BY

## Sommaire

Chapitre 3 - SQL, manipulation de données

9 SQL

16 Renommage - Alias

10 Projections

17 ORDER BY

11 Sélections

18 GROUP BY

12 Fonctions de calcul

19 Ajout de données

13 Opérations arithmétiques

20 Mise à jour de données

14 Dates et Heures

21 Supprimer des données

15 Jointures

F. Gerard (HEPL)

Intro R&B

2014-2015

127 / 140

ORDER BY

## Tri du résultat

```
SELECT * FROM table ORDER BY colonne1 [ASC|DESC] [,colonne2 [ASC|DESC]] ;
```

### Exemples

- SELECT nom FROM eleves WHERE annee = 1 ORDER BY nom ;
- SELECT nom, annee FROM eleves ORDER BY nom, annee ;
- SELECT nom, annee FROM eleves ORDER BY annee, nom ;
- SELECT nom, annee FROM eleves ORDER BY annee DESC, nom ;

F. Gerard (HEPL)

Intro R&B

2014-2015

128 / 140

## GROUP BY

**Sommaire**

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑯ Renommage - Alias

⑩ Projections

⑰ ORDER BY

⑪ Sélections

⑯ GROUP BY

⑫ Fonctions de calcul

⑯ Ajout de données

⑬ Opérations arithmétiques

⑳ Mise à jour de données

⑭ Dates et Heures

㉑ Supprimer des données

Jointures

F. Gerard (HEPL)

Intro R&amp;B

2014-2015

129 / 140

## GROUP BY

**Groupement de ligne**

SELECT \* FROM table Group BY colonne ;

Afficher la moyen de chaque élèves

- SELECT nom, AVG(points) FROM resultats GROUP BY num\_eleve ;

## GROUP BY

## Groupement de ligne

explications

- `SELECT AVG(points) FROM resultats;`  
Calcule la moyenne des points de toute la table *resultats*.
- `SELECT AVG(points) FROM resultats WHERE num_eleve = 1;`  
Calcule la moyenne des points de l'élève numéro 1.
- Pour afficher la moyenne des points pour chaque étudiant, il faudrait exécuter la dernière requête pour chaque valeur de `num_eleve`.

## GROUP BY

## Groupement de ligne

explications

- GROUP BY permet de regrouper des lignes ensembles.
- `SELECT * FROM resultats GROUP BY num_eleve;`  
Affiche la première ligne de chaque groupement.
- `SELECT MAX(points) FROM resultats GROUP BY num_eleve;`  
Affiche pour chaque groupement, la ligne pour laquelle la colonne point est maximum.
- `SELECT AVG(points) FROM resultats GROUP BY num_eleve;`  
Affiche pour chaque groupement, la moyenne des points.

GROUP BY

## Groupement de ligne

```
SELECT * FROM table Group BY colonne HAVING ...
```

Afficher la moyenne des élèves

- `SELECT AVG(points) FROM resultats GROUP BY num_eleve;`

Afficher la moyenne des élèves qui ont moins de < 10/20

- `SELECT AVG(points) FROM resultats GROUP BY num_eleve HAVING AVG(points)<10;`

Ajout de données

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑯ Renommage - Alias

⑩ Projections

⑰ ORDER BY

⑪ Sélections

⑲ GROUP BY

⑫ Fonctions de calcul

⑯ Ajout de données

⑬ Opérations arithmétiques

⑳ Mise à jour de données

⑭ Dates et Heures

㉑ Supprimer des données

⑮ Jointures

Ajout de données

## Ajout de données

```
INSERT INTO table VALUES(col1,col2,...) [,(...)] ;
```

### Ajouter une ligne

- `INSERT INTO eleves VALUES  
(10, 'Dujardin', 'Jean', '1962-10-02', 80, 2);`

### Ajouter deux lignes

- `INSERT INTO eleves VALUES  
(11, 'Turner', 'Will', '1865-08-15', 80, 2),  
(12, 'Sparaoow', 'Jack', '1862-08-15', 80, 2);`

Ajout de données

## Ajout de données

```
INSERT INTO table(col2,col4) VALUES(col2,col4) [,(...)] ;
```

### Exemple

- `INSERT INTO eleves(nom, num_eleve, date_naissance, prenom)  
VALUES('Dujardin', 10, '1962-10-02', 'Jean');`

Mise à jour de données

## Sommaire

Chapitre 3 - SQL, manipulation de données

<a href="#">9 SQL</a>	<a href="#">16 Renommage - Alias</a>
<a href="#">10 Projections</a>	<a href="#">17 ORDER BY</a>
<a href="#">11 Sélections</a>	<a href="#">18 GROUP BY</a>
<a href="#">12 Fonctions de calcul</a>	<a href="#">19 Ajout de données</a>
<a href="#">13 Opérations arithmétiques</a>	<a href="#">20 Mise à jour de données</a>
<a href="#">14 Dates et Heures</a>	<a href="#">21 Supprimer des données</a>

Jointures    F. Gerard (HEPL)    Intro R&B    2014-2015    137 / 140

Mise à jour de données

UPDATE table SET col1=..., col2= ... [WHERE condition] ;

**Exemples**

- UPDATE eleves SET nom='Lagafe', prenom='Gaston' ;
- UPDATE professeurs SET nom='Lagafe' WHERE num\_prof = 1 ;

**Attention !!!**

- Bien définir la condition, car modifie toutes les lignes qui seraient affichées si l'on avait mis SELECT \* FROM ... WHERE ... en lieu et place de UPDATE ... WHERE .... .
- On ne sait pas modifier une clé primaire.

F. Gerard (HEPL)    Intro R&B    2014-2015    138 / 140

Supprimer des données

## Sommaire

Chapitre 3 - SQL, manipulation de données

⑨ SQL

⑯ Renommage - Alias

⑩ Projections

⑭ ORDER BY

⑪ Sélections

⑮ GROUP BY

⑫ Fonctions de calcul

⑯ Ajout de données

⑬ Opérations arithmétiques

⑰ Mise à jour de données

⑭ Dates et Heures

㉑ Supprimer des données

㉒ Jointures

F. Gerard (HEPL)

Intro R&B

2014-2015

139 / 140

Supprimer des données

## Supprimer des données

`DELETE FROM table [WHERE condition];`

### Exemples

- `DELETE FROM eleves ;`
- `DELETE FROM professeurs WHERE num_prof = 1 ;`

### Attention !!!

- Bien définir la condition, car efface toutes les lignes qui seraient affichées si l'on avait mis `SELECT *` en lieu et place de `DELETE`.
- On ne sait pas effacer une ligne qui est référencée dans une autre table (contrainte d'intégrité référentielle).

F. Gerard (HEPL)

Intro R&B

2014-2015

140 / 140

# Bibliographie

- [1] Programmer spip. <http://programmer.spip.net>.
- [2] Spip pour les nuls. <http://spippourlesnuls.fr>.
- [3] Spip.net. [http://www.spip.net/fr\\_rubrique91.html](http://www.spip.net/fr_rubrique91.html).
- [4] Pierre Delmal. *SQL2 - SQL3 Application à ORACLE*. De Boeck Université,  
3<sup>e</sup>édition, 2000.



# Table des matières

<b>I</b>	<b>Introduction</b>	<b>1</b>
<b>1</b>	<b>Sites web statiques VS dynamiques</b>	<b>3</b>
1.1	Sites web statiques . . . . .	3
1.2	Sites web dynamiques . . . . .	3
1.3	Les différents types de sites dynamiques . . . . .	4
1.4	Mise en oeuvre d'un site dynamique . . . . .	5
<b>2</b>	<b>Publication de sites web</b>	<b>7</b>
2.1	Serveur web - Serveur HTTP - Sites web . . . . .	8
2.2	Serveurs dédiés - Serveurs mutualisés . . . . .	10
2.3	URL & DN . . . . .	12
2.4	Sous domaine . . . . .	14
2.5	Choisir un hébergeur . . . . .	16
2.6	Publication sur Internet . . . . .	18
<b>3</b>	<b>Droits d'accès des fichiers et dossiers sous UNIX</b>	<b>25</b>
3.1	User . . . . .	26
3.2	Group . . . . .	26
3.3	Attribution du user/group . . . . .	26
3.4	Droits . . . . .	27
3.5	Partager un dossier . . . . .	27
3.6	Cas concret du serveur Carl . . . . .	30
<b>II</b>	<b>SPIP</b>	<b>33</b>
<b>4</b>	<b>Découverte de SPIP</b>	<b>35</b>
4.1	Présentation de SPIP . . . . .	36

4.1.1	C'est quoi SPIP ? . . . . .	36
4.1.2	Fonctionnement de SPIP . . . . .	36
<b>5</b>	<b>Installation de SPIP</b>	<b>37</b>
5.1	Outils nécessaires . . . . .	38
5.2	Où placer les fichiers de spip . . . . .	38
5.2.1	En local . . . . .	38
5.2.2	En distant . . . . .	38
5.3	Publication et installation de spip . . . . .	39
<b>6</b>	<b>Conception de squelettes</b>	<b>47</b>
6.1	Documentation SPIP . . . . .	48
6.1.1	Boucles et balises . . . . .	48
6.1.2	Critères et filtres . . . . .	48
6.1.3	D'autres pages . . . . .	48
6.2	Les balises propres au site . . . . .	48
6.3	Syntaxe des boucles . . . . .	49
6.3.1	Boucle simple . . . . .	49
6.3.2	Boucle avec critères . . . . .	50
6.3.3	Syntaxe complète des boucles . . . . .	51
6.4	Syntaxe complète des balises . . . . .	51
6.5	Les autres pages . . . . .	52
6.6	La balise #ENV . . . . .	54
6.7	Les filtres . . . . .	54
6.7.1	Filtres sur les dates . . . . .	55
6.7.2	Le retour de if else . . . . .	55
6.8	Secteur et branche . . . . .	57
6.9	Une boucle dans une boucle . . . . .	58
6.10	Compter le nombre d'élément d'une boucle . . . . .	59
6.11	GET & SET . . . . .	59
6.12	404, File not found . . . . .	59
6.13	Faire un menu correctement . . . . .	62
6.13.1	Activer le bon lien . . . . .	62
6.13.2	Trier les liens . . . . .	64
6.13.3	Pages statiques dans le menu . . . . .	64
6.14	#EXPOSE . . . . .	65
6.15	Boucle récursive . . . . .	66
6.16	Les formulaires . . . . .	68
6.16.1	Formulaire de recherche . . . . .	68
6.16.2	Formulaire de forum . . . . .	69
<b>III</b>	<b>SGBD</b>	<b>71</b>
<b>7</b>	<b>Modèle relationnel &amp; SQL</b>	<b>74</b>
7.1	Introduction . . . . .	74
7.2	Le modèle relationnel . . . . .	75
7.2.1	Objets . . . . .	77
7.2.2	Contraintes . . . . .	84
7.2.3	Opérateurs . . . . .	89

<b>8 Comment placer une clé étrangère</b>	<b>94</b>
8.1 0,1 - 0,N . . . . .	95
8.2 1,1 - 0,N . . . . .	98
8.3 0,1 - 0,1 . . . . .	100
8.4 1,1 - 0,1 . . . . .	103
8.5 0,N - 0,N . . . . .	105
8.6 En résumé . . . . .	107
<b>9 SQL, manipulation de données</b>	<b>108</b>
9.1 SQL . . . . .	109
9.2 Projections . . . . .	113
9.3 Sélections . . . . .	115
Condition de base . . . . .	116
BETWEEN . . . . .	117
IN . . . . .	118
LIKE . . . . .	120
NULL . . . . .	121
AND . . . . .	122
OR . . . . .	123
Conditions multiples . . . . .	124
9.4 Fonctions de calcul . . . . .	125
9.5 Opérateurs arithmétiques . . . . .	127
9.6 Dates et Heures . . . . .	128
9.7 Jointures . . . . .	130
Produit cartésien . . . . .	130
Jointres . . . . .	131
Jointure sur 3 tables . . . . .	133
9.8 Renommage - Alias . . . . .	135
9.9 ORDER BY . . . . .	136
9.10 GROUP BY . . . . .	137
9.11 Ajout de données . . . . .	139
9.12 Mise à jour de données . . . . .	141
9.13 Supprimer des données . . . . .	142
<b>Bibliographie</b>	<b>143</b>