

République Algérienne Démocratique Et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab Blida
Faculté des Sciences
Département d'informatique



Mémoire de fin d'étude pour l'obtention du diplôme de Master en informatique
option Génie des systèmes informatiques

Thème

Conception et réalisation d'une application de synchronisation site joomla local - site joomla distant

Promoteur:

Mr. CHERIF ZAHAR Sid-Ahmed Amine

Réalisé par:

Mr. BOUCHOUL Billel

Président: Ferjera Sofiane
: cherfa imane
amer Khadiouja

Année Universitaire

2013/2014

MA-004-228-1

>> Remerciements <<

*Je remercie « Dieu » de m'avoir
accordé les connaissances de la science et de m'avoir
aidé à réaliser ce travail.*

*Au terme de ce modeste travail je tiens
à remercier chaleureusement et respectivement tous ceux qui ont
contribués de près ou de loin à la réalisation
de ce modeste projet de fin d'étude.*

*Mes vifs remerciements vont tous d'abord
à Mr. Cherif Zahar Sid-Ahmed Amine qui ma a aide à développer
notre travail.*

*Je tiens à remercier tous les Enseignants qui ma suivis durant notre
formation.*

Et Merci à Tous...

Dédicaces

*A mes chers Parents,
Que nulle dédicace ne puisse exprimer ce que je leurs dois, pour leur
bienveillance, leur affection et leur soutien...
Trésors de bonté, de générosité et de tendresse, en témoignage
de mon profond amour et ma grande reconnaissance.*

*A mon cher frère et mes chères sœurs,
En témoignage de mes sincères reconnaissances pour les efforts qu'ils ont
consentis pour l'accomplissement de mes études.
Je leur dédie ce modeste travail en témoignage de mon grand amour
et ma gratitude infinie.*

A toute ma Famille

*A tous mes Amis,
Pour leur aide et leur soutien moral durant l'élaboration
du travail de fin d'étude.*

Et A tous le Monde...

Billel

Sommaire

Résumé

Introduction générale.....	1
Problématique	3
Objectif.....	3

PARTIE 1 : ETAT DE L'ART

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

Partie 1 : le gestionnaire de contenu	5
Objectifs.....	5
1. Introduction	5
2. Présentation de CMS Joomla	5
3. Architectures d'un CMS	6
4. Principes	9
5. Caractéristique.....	11
6. Avantages	11
7. Inconvénients	12

Partie 2 : Analyse des incidences sur les tables lors le la mise à jour d'un site joomla.....13

8. Analyse des incidences sur la base de données.....	13
9. Gestion des catégories.	15
9.1.Ajout d'une catégorie	15
9.2.Les incidences sur la base de données lors de l'ajout d'une catégorie.....	16
10. Gestion des articles	18
10.1.Ajout d'un article.....	18
10.2. Les incidences sur la base de données lors de l'ajout d'un article.....	19

11. Gestion des utilisateurs.....	20
11.1 Ajout d'un utilisateur.....	20
11.2. Les incidences sur la base de données lors de l'ajout d'un utilisateur.....	21
12. Gestion des menus.....	23
12.1 Ajout d'un menu.....	23
12.2. Les incidences sur la base de données lors de l'ajout d'un menu.....	23
13. Conclusion.....	25

Chapitre 2 : les types de synchronisation

1. Introduction	26
2. Définition.....	26
3. Principe	27
4. La mise à jour synchrone	27
4.1. La réplication synchrone asymétrique	28
4.2. La réplication synchrone symétrique	29
5. La mise à jour asynchrone	29
5.1. La réplication asynchrone asymétrique	31
5.2. La réplication asynchrone symétrique	31
6. Architecture de la synchronisation/réplication	32
6.1. Maître de plusieurs esclaves	32
6.2. Esclave de deux maîtres	32
6.3. Maître double	32
6.4. Anneau de réplication (multi-maître)	33
6.5. Pyramide	34
7. La synchronisation avec « pro sync framework »	34
7.1. Pourquoi Microsoft Sync Framework?	34
7.2. Principales caractéristiques du cadre de Microsoft Sync	35
7.3. Flux de synchronisation	35

PARTIE 2 : CONCEPTION ET REALISATION

Chapitre 1 : Conception des algorithmes de synchronisation

1. Introduction.	37
2. Les types d'interventions et problèmes qu'ils engendrent pour la synchronisation.....	37
3. Les algorithmes de synchronisation avec suppression.....	39
3.1. Algorithme de synchronisation des catégories entre deux sites joomla.....	39
3.2. Algorithme de synchronisation des articles entre deux sites joomla.....	41
3.3 .Algorithme de synchronisation des utilisateurs entre deux sites joomla.....	43
3.4 .Algorithme de synchronisation des menus entre deux sites joomla.....	45
4. Algorithmes de synchronisation sans suppression.....	47
5. Conclusion.....	50

Chapitre 2 : Réalisation et test

1. Introduction.	51
2. Les outils d'implémentation	51
2.1 VB.....	51
2.2 Wampserver.....	51
2.3 Microsoft Access.....	51
3. Synchronisation avec suppression	51
3.1. Impact sur la base de données du site local.....	53
3.2. Impact sur la base de données du site distant avant la synchronisation.....	56
3.3. Synchronisation	60
3.4 .Impact sur la base de données du site distant après la synchronisation.....	62
3.5 .Impact sur le site distant après la synchronisation.....	66
4. Synchronisation sans suppression	67
4.1. Synchronisation d'un utilisateur.....	72

4.1.1. Cas d'ajout d'un utilisateur	67
4.1.2. Cas de suppression d'un utilisateur	72
4.2. Synchronisation des menus.....	77
4.2.1. Cas d'ajout d'un menu	77
4.2.2. Cas de suppression d'un menu	80
Conclusion général.....	84

La liste des Figures

Figures		Page
Fig : 1	Contenant et contenu	6
Fig : 2	MCD de joomla 1.5	8
Fig : 3	Position de l'interface web d'un CMS	10
Fig : 4	Fonctionnement de CMS	10
Fig : 5	Outil de comparaison et de synchronisation de PMA	14
Fig : 6	Ecran de synchronisation de PMA	16
Fig : 7	Ajout d'une catégorie	15
Fig : 8	Incidences sur la base de données lors de l'ajout d'une catégorie	16
Fig : 9	La table assets après l'ajout d'un catégorie Cat	17
Fig : 10	La table categories après l'ajout d'un catégorie Cat	17
Fig : 11	Ajout d'un article	18
Fig : 12	Incidences sur la base de données lors de l'ajout d'un article	19
Fig : 13	La table assets après l'ajout d'un article Art	19
Fig : 14	La table #_content après l'ajout d'un article Art	20
Fig : 15	Ajout d'un utilisateur	21
Fig : 16	Incidences sur la base de données lors de l'ajout d'un utilisateur	21
Fig : 17	La table users après l'ajout d'un utilisateur billel	22
Fig : 18	La table user_usergroup_map après l'ajout d'un utilisateur billel	22
Fig : 19	Ajout d'un menu	23
Fig : 20	Incidences sur la base de données lors de l'ajout d'un menu	24
Fig : 21	La table menu_types après l'ajout d'un menu menu	24
Fig : 22	La réplication synchrone	28
Fig : 23	La réplication synchrone asymétrique	29
Fig : 24	La réplication synchrone symétrique	29
Fig : 25	La réplication asynchrone	30
Fig : 26	La réplication asynchrone asymétrique	31
Fig : 27	La réplication asynchrone symétrique	31
Fig : 28	maître de plusieurs esclaves	32
Fig : 29	Esclave de deux maîtres	32
Fig : 30	Maître double	33

Fig : 31	Maître double	33
Fig : 32	Anneau de réplication (multi-maître)	33
Fig : 33	Pyramide	34
Fig : 34	Flux de synchronisation	36
Fig : 35	Algorithme de synchronisation des catégories avec suppression	40
Fig : 36	Algorithme de synchronisation des articles avec suppression	42
Fig : 37	Algorithme de synchronisation des utilisateurs avec suppression	44
Fig : 38	Algorithme de synchronisation des menus avec suppression	46
Fig : 39	Algorithme de synchronisation sans suppression	49
Fig : 40	L'interface du site local	52
Fig : 41	L'interface du site distant	52
Fig : 42	La table #_assets après l'ajout des articles (accueil, presentation, services)	53
Fig : 43	La table #_content après l'ajout des articles (accueil, presentation, services)	53
Fig : 44	La table #_assets après l'ajout de la catégorie (Catégorie 1)	54
Fig : 45	La table #_categories après l'ajout de la catégorie (Catégorie 1)	54
Fig : 46	La table #_menu_types après l'ajout d'un menu (menu1)	55
Fig : 47	La table #_users après l'ajout d'un utilisateur (billel)	55
Fig : 48	La table #_user_usergroup_map après l'ajout d'un utilisateur (billel)	56
Fig : 49	La table #_assets du site distant avant la synchronisation des articles	56
Fig : 50	La table #_content du site distant avant la synchronisation des articles	57
Fig : 51	La table #_assets du site distant avant la synchronisation des catégories	57
Fig : 52	La table #_categories du site distant avant la synchronisation des catégories	58
Fig : 53	La table #_users du site distant avant la synchronisation des utilisateurs	58
Fig : 54	La table #_user_usergroup_map du site distant avant la synchronisation des utilisateurs	59
Fig : 55	La table #_menu_types du site distant avant la synchronisation des menus	59
Fig : 56	l'interface de l'application de synchronisation avec suppression	60
Fig : 57	Les boutons de synchronisation des articles	61
Fig : 58	Les boutons de synchronisation des catégories	61
Fig : 59	Les boutons de synchronisation des utilisateurs	61
Fig : 60	Les boutons de synchronisation des menus	62
Fig : 61	La table #_assets du site distant après la synchronisation des articles	62
Fig : 62	La table #_content du site distant après la synchronisation des articles	63

Fig : 63	La table #_assets du site distant après la synchronisation des catégories	63
Fig : 64	La table #_categories du site distant après la synchronisation des catégories	64
Fig : 65	La table #_users du site distant après la synchronisation des utilisateurs	64
Fig : 66	La table #_user_usergroup_map du site distant après la synchronisation des utilisateurs	65
Fig : 67	La table #_menu_types du site distant après la synchronisation des menus	65
Fig : 68	Le site distant après la synchronisation	66
Fig : 69	La table #_users du site maitre après l'ajout d'un utilisateur billel	67
Fig : 70	La table #_user_usergroup_map du site maitre après l'ajout d'un utilisateur billel	68
Fig : 71	La table #_users du site esclave avant la synchronisation	68
Fig : 72	La table #_users du site esclave avant la synchronisation	69
Fig : 73	La table des requêtes SQL des utilisateurs	70
Fig : 74	La table des paramètres des utilisateurs	70
Fig : 75	Bouton de synchronisation sans suppression	71
Fig : 76	La table #_users du site distant après la synchronisation sans suppression	71
Fig : 77	La table #_user_usergroup_map du site distant après la synchronisation sans suppression	72
Fig : 78	La table #_users du site local après la suppression de l'utilisateur (billel)	72
Fig : 79	La table #_user_usergroup_map du site local après la suppression de l'utilisateur (billel)	73
Fig : 80	La table #_users du site distant avant la synchronisation sans suppression	74
Fig : 81	La table #_user_usergroup_map du site distant avant la synchronisation sans suppression	74
Fig : 82	La table des requêtes pour la suppression des utilisateurs	75
Fig : 83	La table des paramètres pour la suppression des utilisateurs	75
Fig : 84	La table #_users après la synchronisation des utilisateurs sans suppression	76
Fig : 85	La table #_user_usergroup_map après la synchronisation des utilisateurs sans suppression	76
Fig : 86	La table #_menu_types après l'ajout d'un menu au site local	77
Fig : 87	La table #_menu_types du site distant avant la synchronisation sans suppression	78
Fig : 88	La table des requêtes pour l'ajout d'un menu	79
Fig : 89	La table des paramètres pour l'ajout d'un menu	79

Fig : 90	La table #_menu_types du site distant après la synchronisation sans suppression	80
Fig : 91	La table #_menu_types apres la suppression du menu	81
Fig : 92	La table #_menu_types du site distant avant la synchronisation	81
Fig : 93	La table des requêtes pour la suppression d'uns menu	82
Fig : 94	La table des paramètres pour la suppression d'uns menu	82
Fig : 95	La table #_menu_types après la synchronisation des menus sans suppression	83

La liste des tableaux

Tableau		Pages
Tab.1 :	Listes des tables de base de données joomla 1.5	7
Tab.2 :	Structure du CMS	9

Résumé

Français

Le présent mémoire traite d'un problème crucial qui se pose aux développements des sites web notamment ceux basés sur Joomla.

Un site web est développé en local. Une fois qu'il a pris l'aspect recherché par le développeur, il est mis en ligne.

Dés lors, le site entré en phase de production il commence à « grossir », aucun changement ne peut plus être fait sauf en ligne.

Alors, il est souvent plus pratique d'apporter des modifications sur le site local. C'est le cas par exemple en l'absence de connexion internet. C'est aussi le cas quand des changements peuvent conduire à un dysfonctionnement du site.

Le présent travail, veille à essayer de proposer une solution pour garder les sites local et distant similaires l'un l'autre facilitant aussi les aussi les retouches en local et distant.

English

This memo addresses a crucial problem to the developments of websites including those based on Joomla.

A website is developed locally. Once he took the sought by the developer aspect, it is posted. Henceforth, the site went into production it begins to "grow", no changes can be made except online.

So, it is often more practical to make changes to the local site. This is the case for example if no Internet connection. it is also the case when changes can lead to dysfunction of the site.

This work ensures try to propose a solution to keep local and remote locations like each other as easy as the touch-ups in local and distant.

العربية

تتناول هذه المذكرة مشكلة هامة التي ترتبط بتطوير مواقع الانترنت بصفة خاصة المواقع المطورة بـ « Joomla ».

بعد الانتهاء من تطوير موقع معين محليا يقوم المطور بوضعه عن بعد على الانترنت من اجل استعماله من قبل المستخدمين العاديين. عندما يصبح الموقع في وضع التشغيل أو الاستعمال يصبح من الصعب القيام بالتغييرات محليا مما يستوجب القيام بالتغييرات عن بعد (على الموزع), مع هذا يمكن القيام بالتغييرات محليا عندما يكون الاتصال بالانترنت معدوما أو عندما يصبح الموقع البعيد غير صالح او يحتوي على بعض المشاكل التي تعيقه عن الاشتغال بشكل جيد .

المشروع الحالي يحاول القيام بالحفاظ على نفس المعلومات بين الموقع المحلي والموقع البعيد وكذلك المحافظة على آخر التغييرات بين الموقع المحلي والموقع البعيد(على الموزع).

Introduction Générale

Introduction générale

Introduction

À partir des années 2000, notre conception du Web a commencé à changer. On a découvert qu'on pouvait en faire bien plus qu'une affiche publicitaire, qu'il était possible de le transformer en un véritable lieu d'échanges où le visiteur ne serait plus seulement lecteur mais aussi acteur. C'était en fait le début d'une véritable révolution du Web, que certains ont nommée « Web 2.0 ».

Les premiers sites informatifs étaient dits statiques car leur contenu ne changeait que très rarement, au bon vouloir de leur créateur, le webmaster. Leurs successeurs, les sites web participatifs, ont été appelés sites dynamiques car leur contenu pouvait être modifié à tout moment par n'importe quel visiteur.

Aujourd'hui, on ne conçoit plus le Web autrement que par ses sites dynamiques qui invitent l'internaute à participer :

- Les blogs : ils sont régulièrement mis à jour par leurs créateurs et les lecteurs peuvent participer en commentant les billets. Ce sont des sites dynamiques ;
- Les forums : ce sont de véritables espaces de discussion en ligne ;
- Les réseaux sociaux : Facebook et Twitter, pour ne citer qu'eux, invitent les internautes à échanger entre eux, que ce soient des messages, des photos, des vidéos... Ce sont eux aussi des sites dynamiques.

En fait, la plupart des sites web d'aujourd'hui sont des sites dynamiques. On pourrait ajouter de nombreux autres sites dans cette catégorie, comme les moteurs de recherche, les webmails, les wikis, etc.

La création d'un site dynamique offre beaucoup d'avantage par rapport à un site statique: mise à jour tant sur le fond que sur la forme, intégration de fonctions de personnalisation, souplesse, facilité de maintenance des données.

Plus un site statique devient important, plus son actualisation devient difficile, alors que dans le cas d'un site dynamique la gestion des données sont réalisée directement dans une base de données (SQL MySQL).

Parmi les avantages on peut citer :

Introduction générale

- Possibilité d'ajouter, de modifier ou d'effacer le contenu de votre site en direct Transformation instantanée quand vous le souhaitez sans formation préalable.
- Les clients ou des internautes visiteurs peuvent voter, témoigner, donner leurs avis sur un de vos produits ou sur vos services.
- Possibilité de limiter l'accès de votre site ou d'une partie par le biais de nom et mot de passe (ex : créer un espace réservé à pour les clients).
- Optimisation de l'interactivité spécifique à Internet.
- Création d'une base de données exploitable.

Mais il y a des inconvénients :

- Moins rapide qu'un site statique
- la consommation de ressources serveur est plus grande,
- présente une plus grande complexité,
- à une moindre sécurité,
- Le problème de mise à jour. Une fois que le site rentre en phase de production, il devient impossible de continuer à opérer des changements sur le site local. Elles n'auraient aucun impact. Or, on peut se trouver dans des situations où on voudrait continuer à faire des modifications localement tout en ayant le site en ligne. Le problème qui va se poser alors : comment répercuter les changements sans dommages.

Introduction générale

Problématique

La plateforme Joomla est très utilisée pour le développement de site web de types gestionnaires de contenus.

Un site Joomla contient une base de donnée MySQL qui se trouve sur internet, Ce site doit être tout le temps fonctionnel.

Pour parer les éventuelles pertes de données, on se trouve dans l'obligation de faire des sauvegardes périodiques, de les télécharger et de les protéger.

Lorsqu'un site prend de la taille, les sauvegardes deviennent lourdes, leur téléchargement prend alors du temps.

Les procédures de récupération ne prendront pas en compte les dernières modifications. Ajouter à cela que, dans des régions où la connexion à internet peut être défaillante, on aimerait bien continuer à travailler localement avec possibilité de répercuté automatiquement les changements.

Pour avoir un site local qui soit le miroir d'un site en ligne -garantissant ainsi protection et possibilité d'agit localement ou en ligne- un dispositif de synchronisation des bases locales et distantes s'avère être nécessaire.

Objectifs

L'objectif principal de notre travail consiste à réaliser la synchronisation des bases locales et distantes, particulièrement en ce qui concerne le contenu : articles, menus, catégories,...

D'autres synchronisations comme celles des trempâtes, plugins etc. ne font pas l'objet de cette étude.

Introduction générale

Organisation de mémoire :

Afin de répondre à la problématique présentée précédemment, ce mémoire est organisé comme suit :

PARTIE 1 : ETAT DE L'ART

Elle est consacrée pour les différents concepts liés à cette étude.

Chapitre 1 : Les questionnaires de contenu.

Chapitre 2 : Analyse des incidences sur les tables lors de la mise à jour d'un site Joomla.

PARTIE 2 : CONCEPTION ET REALISATION

Chapitre 1 : Conception des algorithmes de synchronisation.

Chapitre 2 : Réalisation et test.

Enfin une conclusion générale pour évaluer objectivement le travail et parler des perspectives pour le développer plus en profondeur.

Chapitre 1

Les CMS et le problème de maintenance
de site à jour localement et à distance

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

Objectif

Dans cette partie nous allons introduire les concepts de base des CMS, leurs architectures, caractéristiques générales, et les avantages et inconvénient.

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

1 .Introduction

De nos jours un site web constitue l'une des meilleures façons de se faire connaître au grand publique que l'ont soit un particulier ou un organisme.

Il s'agit d'un moyen de diffusion plus efficace et plus économique que les méthodes traditionnelles telles que les brochures, prospectus, dépliants...etc. sur la quel on peut aisément apporter des mises à jour afin de suivre l'évolution des nouveaux besoins.

En outre, le site web d'une organisation représente, aujourd'hui, son premier point d'accès avec l'extérieur. Il doit fournir des informations de qualité, pertinentes et actualisées à ses visiteurs. La gestion de contenu offre la possibilité de gérer en équipe des sites web à grande échelle d'une manière simple et efficace. Cette technique est devenue un facteur clef de la stratégie web de toute organisation.

Partie 1 : le gestionnaire de contenu

2.Présentation de CMS Joomla

Joomla! est un système libre et gratuit pour la création de sites Web. Il s'agit d'un projet Open source qui, comme la plupart des projets Open Source, est en évolution constante. Il connaît un large succès depuis maintenant plus de sept ans et est l'un des plus populaires avec des millions d'utilisateurs à travers le monde. [DOC_01]

Le terme "Joomla" est dérivé du mot Jumla signifiant en langage Swahili "tous ensemble». Le projet Joomla! est le fruit d'une discussion animée entre la Fondation Mambo, fondée en août 2005, et ses équipes de développement. Joomla! est en effet un prolongement du système Mambo. Joomla! Est utilisé dans le monde entier, que ce soit pour de simples pages d'accueil comme pour des sites plus complexes de grandes entreprises. Il est facile à installer, à administrer et il est très fiable. [DOC_01]

Tout au long de ces sept dernières années, l'équipe de Joomla! n'a eu de cesse de se réorganiser afin de répondre au mieux aux demandes grandissantes des utilisateurs.

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

Joomla est l'un des derniers efficaces systèmes de gestion de contenu open source et des services de développement sur le web qui est écrit avec le langage de script PHP et utilise MySQL comme base de données. Joomla est utilisé pour développer toutes sortes d'applications Web. Joomla dispose d'une structure souple, sûre et modulaire. Il est également facile à utiliser, parcourir et gérer toutes les fonctionnalités du site.

3. Architecture d'un CMS

3.1. Utilisation d'une interface Web classique

Les logiciels de gestion de contenu Web utilisent comme interface un navigateur Web classique. Les navigateurs Web offrent l'avantage de fonctionner dans tous les environnements et sur toutes les plateformes informatiques. [DOC_02]

Ils ne nécessitent pas l'installation de logiciels spécifiques. De plus, la convivialité des navigateurs permet une prise en main rapide des outils de CMS. Aujourd'hui, Internet Explorer, Mozilla, Opéra et Safari sont les navigateurs Web les plus couramment utilisés. [DOC_02]

3.2. Utilisation de gabarits et de feuilles de styles

Un système de gestion de contenu utilise des gabarits pour réaliser des pages dynamiques. Un gabarit permet de réaliser une page modèle qui sera utilisée pour travailler indépendamment sur le contenu ou la forme. En règle générale, la présentation du contenu (polices, couleurs, tailles...) fait appel à des feuilles de styles qui permettent l'obtention d'une présentation rapide, unifiée et automatique.

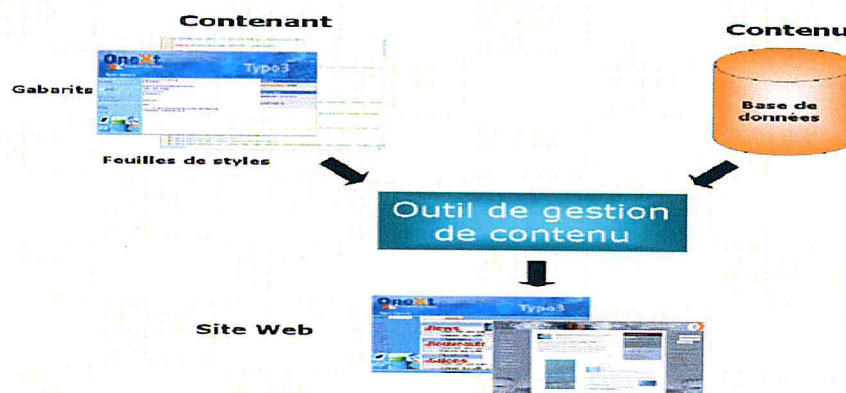


Figure 1 : Contenant et contenu

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

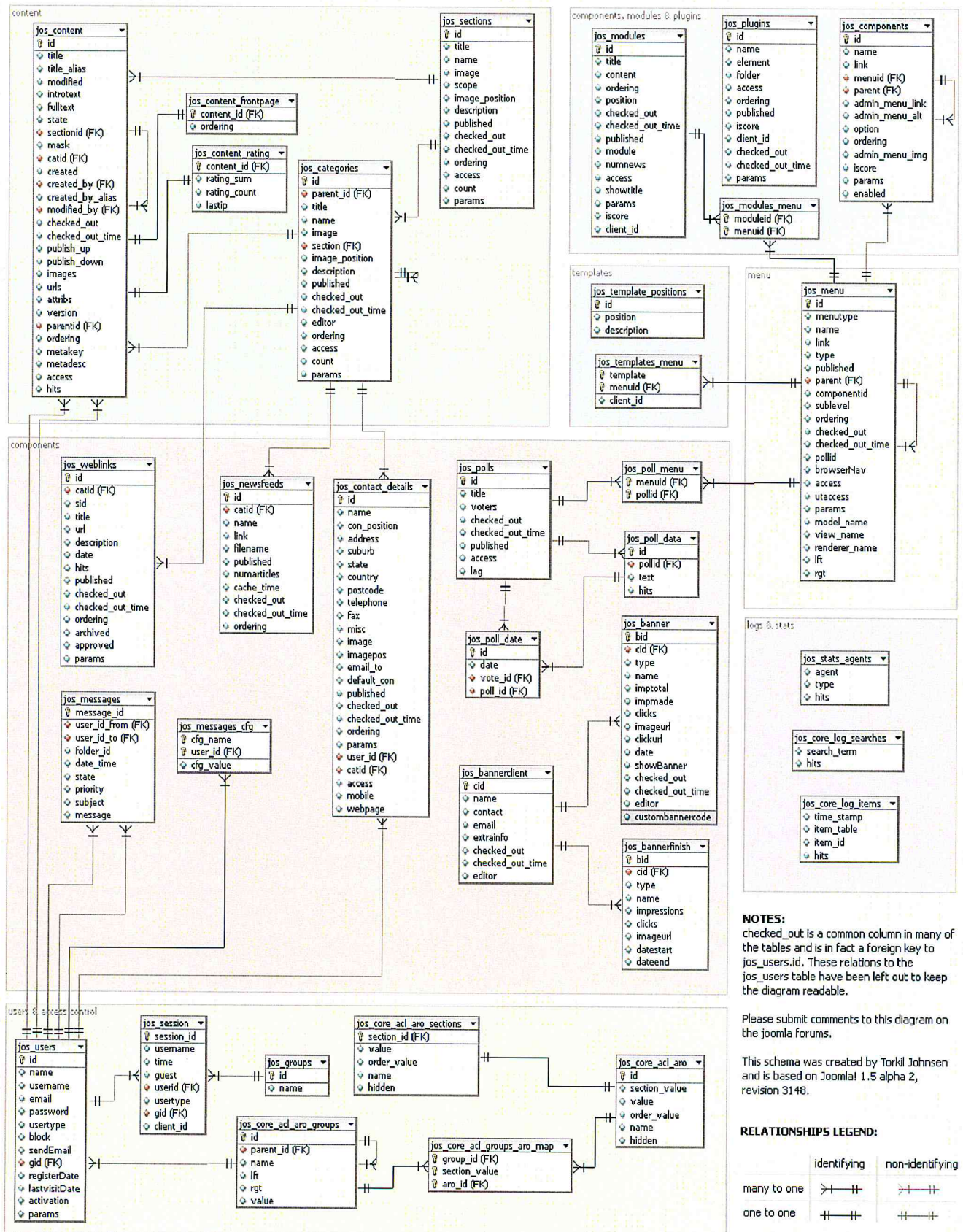


Figure 2 : MCD de joomla 1.5 [WEB_O7]

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

4. Principes

4.1. Front Office

Le terme Front Office est terme d'architecture logicielle. Il désigne la partie qui prend en charge l'interface d'une application, par opposition du back office qui lui regroupe la partie gestion.

On retrouve typiquement le front office dans les sites web commerciaux qui permettent de commander en ligne. Dans ce cas, le front office correspond à la partie hypertexte (web) de l'application, donc ce qui est visible par le consommateur. Elle lui permet d'interagir avec le reste du site. [DOC_02]

4.2. Back Office

Le back office est une partie d'un site web ou d'un système informatique. Elle concerne la partie qui permet d'administrer et de gérer le site. On pourra retrouver comme service par exemple :

- Modification de paramètres.
- Administration d'un forum.
- Gestion des utilisateurs.

CMS	Front Office	Back Office
<p>The diagram illustrates the structure of a CMS. It is divided into two main sections: 'Public' (Front Office) and 'Privé' (Back Office). The 'Public' section contains 'Pages du site' and is accessed via an 'Interface web'. The 'Privé' section contains 'Contenus (données)', 'Mises en forme (squelettes)', and a 'Modèle éditorial' (including typology, rules, and classifications). Both sections are accessed via a standard browser. Below the diagram, 'Visiteurs' are shown interacting with the Front Office, and 'Administrateurs et rédacteurs' are shown interacting with the Back Office.</p>	<p>Partie <<publique>> du site internet géré par le CMS</p> <p>Le contenu est mis en page et présenté au visiteur du site</p>	<p>Partie <<Privé>> nécessitant une authentification</p> <p>Interface permettant la gestion du contenu par les utilisateurs autorisés</p>

Tableau 2: Structure du CMS

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

4.3. Utilisation d'interfaces Web

L'accès au Back Office se fait via un navigateur Internet classique et la gestion du contenu se fait à l'aide de formulaires Web .Ces formulaires une fois validés sont traités par le serveur qui enregistre et organise le contenu en conséquence. [DOC_02]

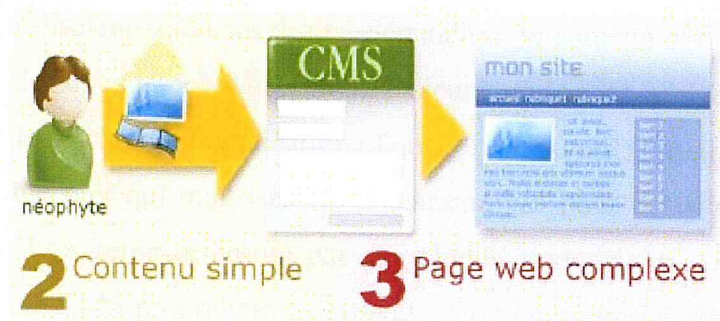


Figure 3 : position de l'interface web d'un CMS

4.4. Séparation contenu/présentation

Le contenu est stocké dans une base de données mais la structure et la présentation sont définies dans des fichiers physiques de type : (modèles, templates, gabarits, squelettes...).

Lors de la consultation d'une page, le CMS intègre le contenu dans le modèle de page et affiche le résultat au visiteur et le système de mise en cache permet de ne pas calculer la page à chaque affichage.

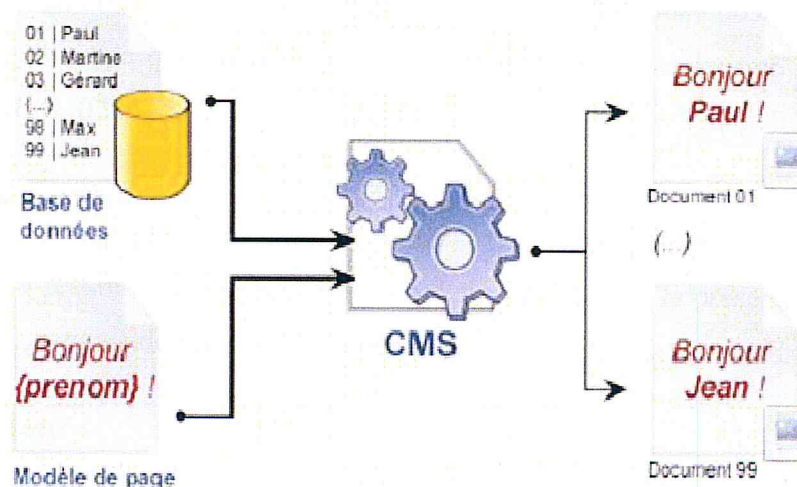


Figure 4 : fonctionnement de CMS

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

Partie 2 : Analyse des incidences sur les tables lors de la mise à jour d'un site joomla

8. analyse des incidences sur la base de données

La mise à jour d'un site joomla nécessite des changements sur la base de données, pour voir les changements dans cette base de données, on va créer un autre site ensuite, on compare la base de données du premier site avec la base de données du deuxième site.

La première base c'est la base source sur laquelle des changements ont été effectués et la deuxième c'est la base cible qui contient l'état initial de la base du site.

Pour faire la comparaison on utilise phpmyadmin (PMA est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.)

Une caractéristique étonnante inclus dans PMA est la possibilité de comparer les bases de données, il permet de comparer une base de données de production à distance (en cours d'exécution sur un hôte Web, par exemple) à une base de données locale (peut-être pour le développement), ou vice versa. **[DOC _04], page 204**

La comparaison d'une base de données est aussi simple que la mise en place de la source et cible de la comparaison. Suivons ces étapes:

1. revenons à l'écran d'accueil en PMA et cliquons sur l'onglet Synchroniser.
2. Lorsque nous cliquons sur l'onglet Synchroniser, la forme de synchronisation s'affiche, comme indiqué dans Figure 5 Ce premier écran permet de définir les informations d'accès pour les deux bases de données pour être comparée et synchronisée.
3. on cliquant sur le bouton exécuter pour commencer la comparaison.
4. l'écran de synchronisation affiche un tableau indiquant la source d'information dans la colonne de gauche et les informations de cible dans la colonne de droite. La colonne du milieu contient des boutons des différences pour S et D qui signifient Structure et données comme indiquer dans la **Figure 6**.

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

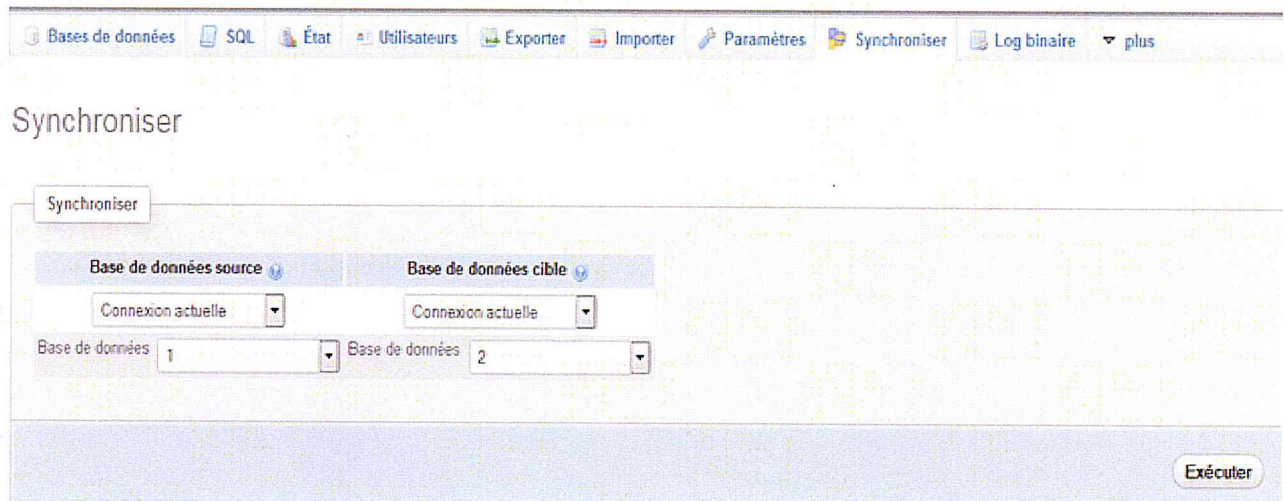


Figure 5 : Outil de comparaison et de synchronisation de PMA

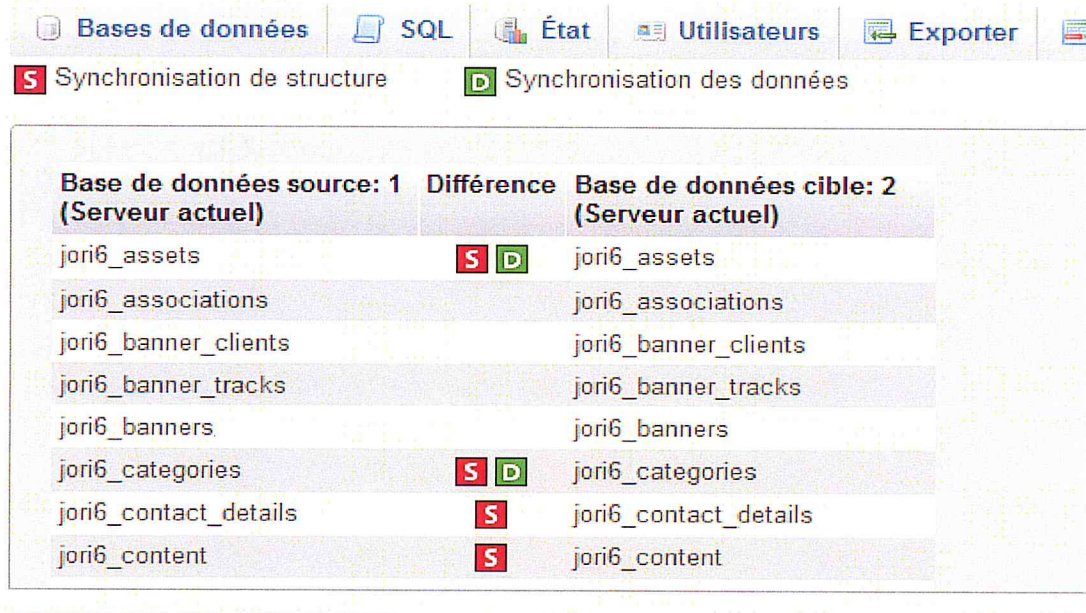


Figure 6 : Ecran de synchronisation de PMA

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

9. Gestion des catégories

En général, la catégorisation est le processus pour reconnaître, différencier et de comprendre quelque chose au moyen de l'abstraction. Cela peut paraître compliqué, mais s'avère très utile lors de la gestion de nombreux articles. [DOC_01]

Avec Joomla II est possible de construire des catégories imbriquées puis de lier des articles à n'importe laquelle de ces catégories.

La catégorisation permet de l'arrangement des articles, cette catégorisation il a deux utilité :

1. Pour les rédacteurs cette classification va permettre d'organiser les articles dans des catégories et de filtrer les articles selon tel ou tel catégorie.
2. pour les visiteur les catégories vont permettre l'affichage de liste d'articles appartenant a tel ou tel catégorie ,Par défaut a l'installation de Joomla il ya une catégorie qui est créer qui s'appel non catégorisé.

9.1. Ajout d'une catégorie

Pour la création d'une catégorie revenons sur le panneau d'administration du site Joomla et rendre dans le menu contenu gestion des catégories ajouter une catégorie

On donne un titre de catégorie et on enregistre **Figure 7**

The screenshot shows the Joomla! administration interface for adding a new category. The page title is "Gestion des catégories : Modifier une catégorie". A confirmation message "Catégorie enregistrée" is displayed at the top. The form contains the following fields:

Titre	Cat
Alias	cat
Parent	- Pas de parent -
Statut	Publié
Accès	Accès Public
Droits	Définir les droits
Langue	Toutes
Id	10
Description	

Figure 7 : Ajout d'une catégorie

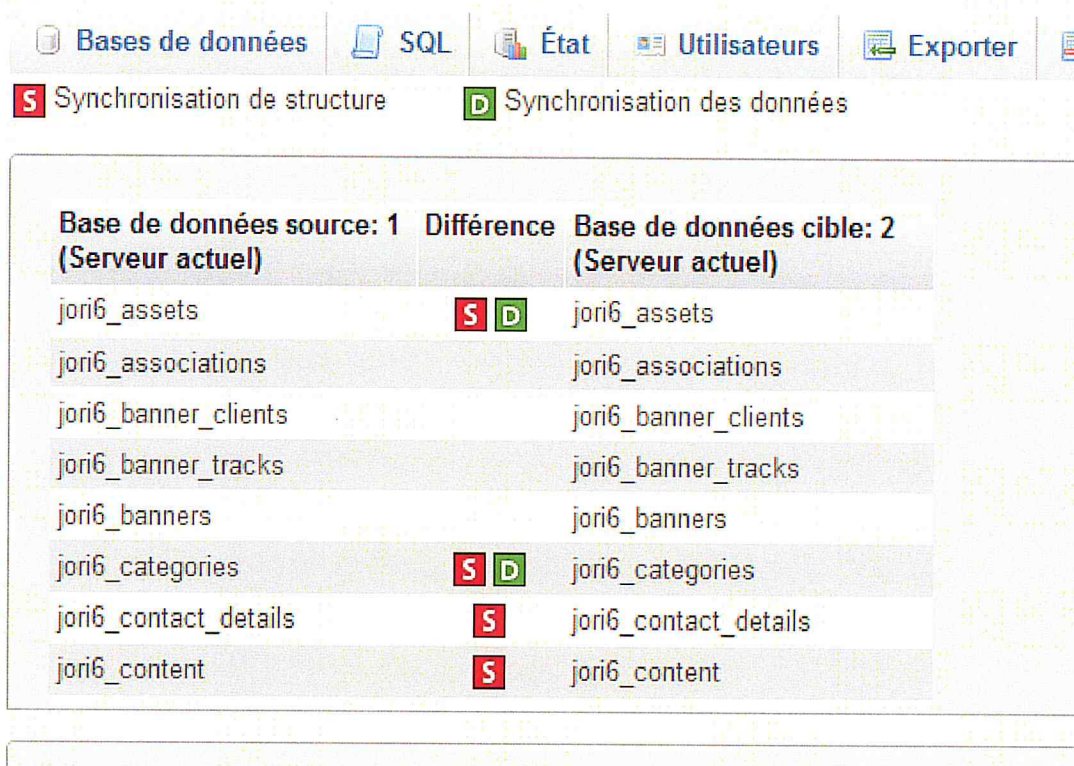
Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

Cette catégorie va enregistrer sur la base de donnée de joomla et implique des incidences sur les tables de la base de donnée joomla.

9.2. Les incidences sur la base de données lors de l'ajout d'une catégorie

Après l'ajout d'une catégorie on fait la comparaison entre la base source et la base cible.

L'écran de synchronisation affiche **Figure 8**.



The screenshot shows a database synchronization tool interface. At the top, there are navigation buttons: 'Bases de données', 'SQL', 'État', 'Utilisateurs', 'Exporter', and a list icon. Below these are two main options: 'S Synchronisation de structure' (highlighted in red) and 'D Synchronisation des données' (highlighted in green). The main area displays a comparison table between two databases.

Base de données source: 1 (Serveur actuel)	Différence	Base de données cible: 2 (Serveur actuel)
jori6_assets	S D	jori6_assets
jori6_associations		jori6_associations
jori6_banner_clients		jori6_banner_clients
jori6_banner_tracks		jori6_banner_tracks
jori6_banners		jori6_banners
jori6_categories	S D	jori6_categories
jori6_contact_details	S	jori6_contact_details
jori6_content	S	jori6_content

Figure 8 : Incidences sur la base de données lors de l'ajout d'une catégorie

Signifie que lors de l'ajout d'une catégorie cette catégorie va enregistrer dans les tables :#_assets et #_categories comme indiquer sur la **Figure 9** et **Figure 10**

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

ID	parent_id	name	type	content
30	19	48	49	2 com_newsfeeds.category.5 Non catégorisé [{"core.create": [], "core.delete": [], "core.edit": []}...
31	25	64	65	2 com_weblinks.category.6 Non catégorisé [{"core.create": [], "core.delete": [], "core.edit": []}...
32	24	60	61	1 com_users.category.7 Non catégorisé [{"core.create": [], "core.delete": [], "core.edit": []}...
33	1	69	70	1 com_finder com_finder [{"core.admin": [{"7": 1}, {"6": 1}], "core.manage": [{"6": 1}]}]
34	1	71	72	1 com_joomlaupdate com_joomlaupdate [{"core.admin": [], "core.manage": [], "core.delete": []}...
35	8	20	21	2 com_content.category.8 a [{"core.create": [{"6": 1, "3": 1}, {"6": 1}]}]
36	8	22	23	2 com_content.category.9 b [{"core.create": [{"6": 1, "3": 1}, {"6": 1}]}]
37	8	24	25	2 com_content.category.10 Cat [{"core.create": [{"6": 1, "3": 1}, {"6": 1}]}]

Figure 9 : La table #_assets après l'ajout d'une catégorie Cat

ID	name	type	content
2	27	1	1 2 1 non-categorise com_content Non catégorisé non-categorise
3	28	1	3 4 1 non-categorise com_banners Non catégorisé non-categorise
4	29	1	5 6 1 non-categorise com_contact Non catégorisé non-categorise
5	30	1	7 8 1 non-categorise com_newsfeeds Non catégorisé non-categorise
6	31	1	9 10 1 non-categorise com_weblinks Non catégorisé non-categorise
7	32	1	11 12 1 non-categorise com_users Non catégorisé non-categorise
8	35	1	13 14 1 a com_content a a
9	36	1	15 16 1 b com_content b b
10	37	1	17 18 1 cat com_content Cat cat

Figure 10 : La table #_categories après l'ajout d'une catégorie Cat

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

10. Gestion des articles :

Dans Joomla un Article est un morceau de contenu composé de texte (HTML), avec éventuellement des liens vers d'autres ressources (par exemple des images). Les articles sont les unités de base de l'information dans le système de contenu et le plus bas niveau dans la hiérarchie du contenu. Depuis Joomla! 2.5, chaque Article est inclus dans une et une seule Catégorie. Une Catégorie peut être elle même incluse dans une autre Catégorie devenant ainsi une sous-catégorie. Il est également possible d'avoir des Articles Non-Catégorisés. Ces articles existent sans être associés à une Catégorie. [WEB_02]

10.1. Ajout d'un article

Pour l'ajout d'un article revenons sur le panneau d'administration du site joomla et rendre dans le menu contenu gestion des articlesajouter un article

On donne un titre et un contenu a cet article et on enregistre **Figure 11**

The screenshot shows the Joomla! Administration interface. At the top, there is a navigation menu with 'Administration' highlighted. Below it, a sub-menu includes 'Site', 'Utilisateurs', 'Menus', 'Contenu', 'Composants', 'Extensions', and 'Aide'. The main content area is titled 'Gestion des articles : Modifier un article' and features an 'Enregistrer' button with a checkmark icon. A blue message bar indicates 'Article enregistré avec succès'. The 'Modifier l'article' form contains the following fields: 'Titre' (Art), 'Alias' (art), 'Catégorie' (- Non catégorisé), 'Statut' (Publié), 'Accès' (Accès Public), 'Droits' (Définir les droits), 'En vedette' (Non), 'Langue' (Toutes), and 'Id' (1). A sidebar on the right lists various options: 'Option', 'Créé par', 'Nom de', 'Date de', 'Début de', 'Fin de pt', 'Révision', and two 'Param' entries.

Figure 11 : Ajout d'un article

Cet article va enregistrer sur la base de donnée de joomla et implique des incidences sur les tables de la base de donnée joomla.

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

10.2. Les incidences sur la base de données lors de l'ajout d'un article

Après l'ajout d'un article on fait la comparaison entre la base source et la base cible.

L'écran de synchronisation affiche **Figure 12**.

Base de données source: 3 (Serveur actuel)	Différence	Base de données cible: 4 (Serveur actuel)
t8nsh_assets	S D	t8nsh_assets
t8nsh_associations		t8nsh_associations
t8nsh_banner_clients		t8nsh_banner_clients
t8nsh_banner_tracks		t8nsh_banner_tracks
t8nsh_banners		t8nsh_banners
t8nsh_categories	S	t8nsh_categories
t8nsh_contact_details	S	t8nsh_contact_details
t8nsh_content	S D	t8nsh_content
t8nsh_content frontpage		t8nsh_content frontpage

Figure 12 : Incidences sur la base de données lors de l'ajout d'un article

Signifie que lors de l'ajout d'une catégorie cette catégorie va enregistrer dans les tables :#_assets et #_content comme indiquer sur la **Figure 13** et **Figure 14**

Afficher	Structure	SQL	Rechercher	Insérer	Exporter	Importer	Opérations	Déclencheurs
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	26	1	63	64	1	com_wrapper	com_wrapper	{}
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	27	8	18	21	2	com_content.category.2	Non catégorisé	{"core.create": [], "core.delete": [], "core.edit": []...
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	28	3	4	5	2	com_banners.category.3	Non catégorisé	{"core.create": [], "core.delete": [], "core.edit": []...
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	29	7	14	15	2	com_contact.category.4	Non catégorisé	{"core.create": [], "core.delete": [], "core.edit": []...
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	30	19	44	45	2	com_newsfeeds.category.5	Non catégorisé	{"core.create": [], "core.delete": [], "core.edit": []...
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	31	25	60	61	2	com_weblinks.category.6	Non catégorisé	{"core.create": [], "core.delete": [], "core.edit": []...
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	32	24	56	57	1	com_users.category.7	Non catégorisé	{"core.create": [], "core.delete": [], "core.edit": []...
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	33	1	65	66	1	com_finder	com_finder	{"core.admin": {"7": 1}, "core.manage": {"6": 1}}
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	34	1	67	68	1	com_joomlaupdate	com_joomlaupdate	{"core.admin": [], "core.manage": [], "core.delete": []}
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	35	27	19	20	3	com_content.article.1	Art	{"core.delete": {"6": 1}, "core.edit": {"6": 1, "4": 1}, "..."}

Figure 13 : La table #_assets après l'ajout d'un article Art

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

The screenshot shows the Joomla! database management interface. At the top, there is a navigation bar with buttons for 'Afficher', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', 'Opérations', and 'Déclencheurs'. Below this, there are options for 'Profilage', 'En ligne', 'Modifier', 'Expliquer SQL', and 'Créer source'. The main area displays a table with columns: id, asset_id, title, alias, title_alias, introtext, fulltext, state, sectionid, mask,catid, created, and created_by. The table contains one row with the following data: id=1, asset_id=35, title=Art, alias=art, introtext='<p>Le Lorem Ipsum est simplement...', fulltext='', state=1, sectionid=0, mask=0, catid=2, created='2014-09-13 16:14:13', created_by=648. Below the table, there are options for 'Tout cocher / Tout décocher', 'Modifier', 'Effacer', and 'Exporter'. At the bottom, there are display options for 'Afficher' with fields for 'Ligne de départ', 'Nombre de lignes', and 'En-têtes à intervalle de'.

id	asset_id	title	alias	title_alias	introtext	fulltext	state	sectionid	mask	catid	created	created_by	cre
1	35	Art	art		<p>Le Lorem Ipsum est simplement...		1	0	0	2	2014-09-13 16:14:13	648	

Figure 14 : La table #_content après l'ajout d'un article Art

11. Gestion des utilisateurs

On peut le dire: la gestion des droits utilisateurs constituent l'une des grandes lacunes de Joomla!. La raison est simple. Joomla! Propose **trois niveaux d'accès** et sept **groupes d'utilisateurs** prédéfinis. Nous ne pourrons pas les modifier, les supprimer ou en créer de nouveaux.

11.1 Ajout d'un utilisateur

Pour l'ajout d'un utilisateur revenons sur le panneau d'administration du site joomla et rendre dans le menu utilisateurs Gestion des utilisateurs Ajouter un utilisateur

On donne un non un identifiant un mot de passe avec confirmation et une adresse de boite email à ce groupe et on enregistre Figure 15

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance



Figure 15 : Ajout d'un utilisateur

Cette utilisateur va enregistrer sur la base de donnée de joomla et implique des incidences sur les tables de la base de donnée joomla.

11.2. Les incidences sur la base de données lors de l'ajout d'un utilisateur

Après l'ajout d'un utilisateur on fait la comparaison entre la base source et la base cible.

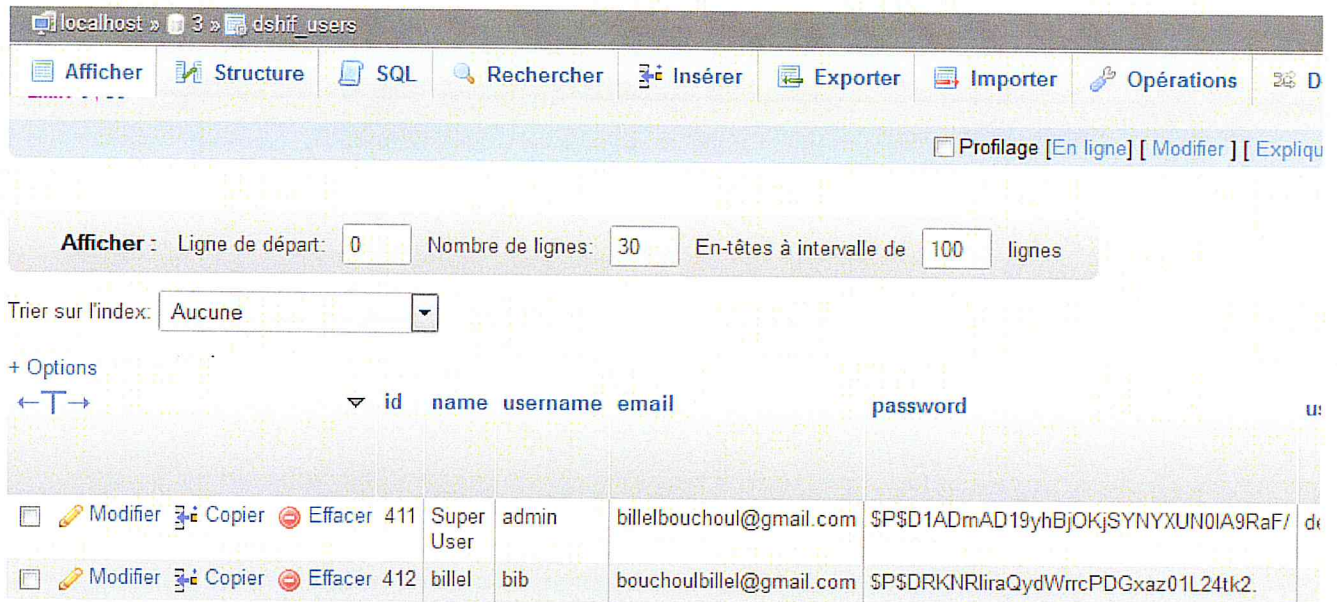
L'écran de synchronisation affiche **Figure 16**

Source	Différence	Cible
dshif_update_sites		dshif_update_sites
dshif_update_sites_extensions		dshif_update_sites_extensions
dshif_updates		dshif_updates
dshif_user_notes		dshif_user_notes
dshif_user_profiles	S	dshif_user_profiles
dshif_user_usergroup_map	D	dshif_user_usergroup_map
dshif_usergroups	S	dshif_usergroups
dshif_users	D	dshif_users
dshif_viewlevels		dshif_viewlevels
dshif_weblinks	S	dshif_weblinks

Figure 16 : Incidences sur la base de données lors de l'ajout d'un utilisateur

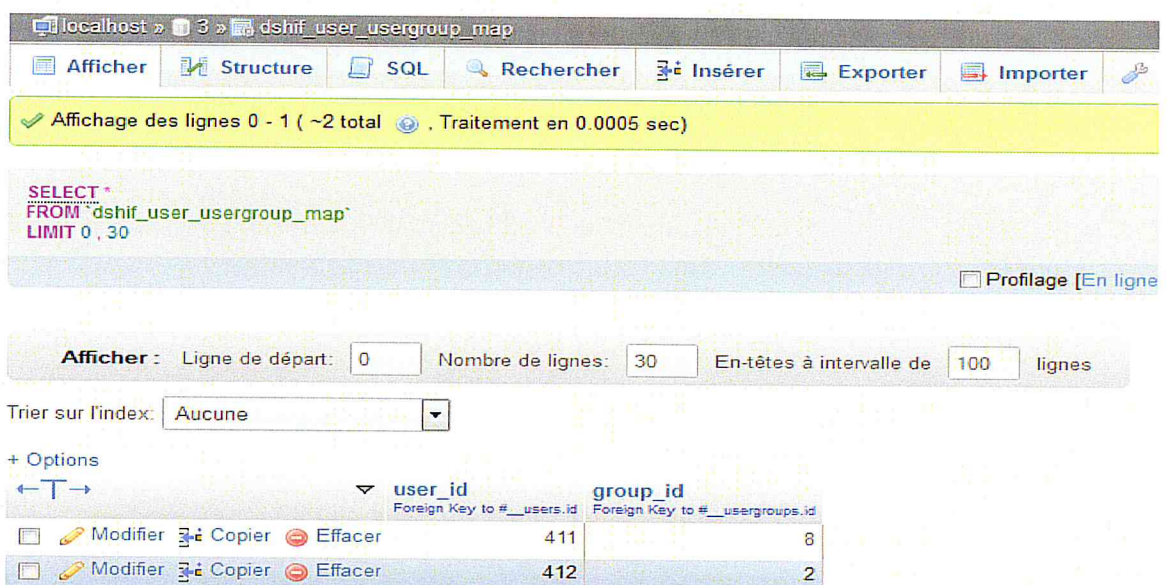
Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

Signifie que lors de l'ajout d'un utilisateur cette utilisateur va enregistrer dans les tables :#_user_usergroup_map et #_users comme indiquer sur la **Figure 17** et **Figure 18**.



	id	name	username	email	password	u:
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	411	Super User	admin	billelbouchoul@gmail.com	SP\$D1ADmAD19yhBJOKjSYNYXUN0IA9RaF/	dr
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	412	billel	bib	bouchoulbillel@gmail.com	SP\$DRKNRliraQydWrrcPDGxaz01L24tk2.	

Figure 17: La table #_users après l'ajout d'un utilisateur **billel**



```
SELECT *
FROM `dshif_user_usergroup_map`
LIMIT 0 , 30
```

	user_id	group_id
	Foreign Key to #_users.id	Foreign Key to #_usergroups.id
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	411	8
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	412	2

Figure 18: La table #_user_usergroup_map après l'ajout d'un utilisateur **billel**


Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

12. Gestion des menus

12.1. Ajout d'un menu

Pour l'ajout d'un menu revenons sur le panneau d'administration du site Joomla et rendre dans le menu Menu Gestion des menus Ajouter un menu

On donne un titre un type et une description à ce menu et on enregistre **Figure 19**



The screenshot shows the Joomla! Administration interface. At the top, there is a navigation bar with the following tabs: Site, Utilisateurs, Menus, Contenu, Composants, Extensions, and Aide. Below the navigation bar, the main heading is 'Gestion des menus : Modifier le menu'. Underneath, there is a section titled 'Menu enregistré' with an information icon. The main content area is titled 'Détails du menu' and contains a form with the following fields:

Détails du menu	
Titre *	menu
Type de menu *	m1
Description	mon premier menu

Figure 19: Ajout d'un menu

Ce menu va enregistrer sur la base de donnée de Joomla et implique des incidences sur les tables de la base de donnée Joomla.

12.2. Les incidences sur la base de données lors de l'ajout d'un menu :

Après l'ajout d'un menu on fait la comparaison entre la base source et la base cible.

L'écran de synchronisation **Figure 20**.

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

dshif_finder_tokens		dshif_finder_tokens
dshif_finder_tokens_aggregate		dshif_finder_tokens_aggregate
dshif_finder_types		dshif_finder_types
dshif_languages		dshif_languages
dshif_menu	S	dshif_menu
dshif_menu_types	D	dshif_menu_types
dshif_messages	S	dshif_messages
dshif_messages_cfg	S	dshif_messages_cfg
dshif_modules	S	dshif_modules
dshif_modules_menu		dshif_modules_menu
dshif_newsfeeds		dshif_newsfeeds

Figure 20 : Incidences sur la base de données lors de l'ajout d'un menu

Signifie que lors de l'ajout d'un menu ce menu va enregistrer dans les tables #_menu_types comme indiquer sur la **Figure 21**

The screenshot shows a database management interface for the table 'dshif_menu_types'. The SQL query displayed is 'SELECT * FROM `dshif_menu_types` LIMIT 0, 30'. Below the query, the table structure is shown with columns: id, menutype, title, and description. The table contains two rows of data:

	id	menutype	title	description
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	mainmenu	Menu principal	Le menu principal du site
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	m1	menu	mon premier menu

Figure 21: La table #_menu_types après l'ajout d'un menu menu

Chapitre 1 : Les CMS et le problème de maintenance de site a jour localement et a distance

13. Conclusion

Dans ce chapitre on a fait une analyse des incidences sur la base de donnée lors de l'ajout d'un catégorie , l'ajout d'un article, l'ajout d'un utilisateur et l'ajout d'un menu pour parvenir a offrir un algorithme permettant de faire la synchronisation.

chapitre 2

Les types de synchronisation

Chapitre 2 : Les types de synchronisation

1. Introduction

Etre capable de faire de la haute disponibilité de machines est une chose qui présente de nombreux avantages. Malheureusement elle ne résout pas à elle seule la disponibilité des services hébergés, et la solution peut parfois être fort complexe dans le cadre des bases de données. Ceci est dû au fait que, contrairement à un serveur web, un site ftp ou tout autre service, il s'agit de gérer un jeu de données qui peuvent être modifiées à fréquence élevée.

2. Définition

Dans un environnement multiutilisateur, la synchronisation est la coordination des accès simultanés à une base de données. En mode synchronisation pessimiste, le système élimine le plus tôt possible les conflits entre les transactions exécutées en parallèle. En mode synchronisation optimiste, les transactions conflictuelles sont annulées le plus tard possible. [DOC_05]

Une transaction est une suite d'opérations caractérisée par les propriétés d'atomicité, de cohérence, d'isolation et de durabilité. La gestion des transactions permet à plusieurs utilisateurs simultanés de travailler sans conflit dans un système de bases de données. [DOC_05]

Une application de base de données repose sur un modèle client-serveur. Dans ce modèle, le client se connecte au SGBD pour passer des ordres. Ces ordres sont de deux natures : lecture (on parle alors de requêtes) ou mise à jour (on parle alors de transactions). Pour les transactions il y a une modification des données sur le serveur, mais cela reste des ordres de courte durée. A l'inverse, dans le cas d'une lecture, il n'y a pas de modification des données mais les traitements peuvent être longs et porter sur une grande masse de données. On comprend donc aisément que, dans le cadre d'un site web par exemple, un nombre important de requêtes peut emboîser partiellement (ou complètement) le serveur. Il existe plusieurs solutions pour palier à ce genre de problèmes et, ça tombe bien). [WEB_O6]

L'objectif principal de la réplication est de faciliter l'accès aux données en en augmentant la disponibilité. Soit parce que les données sont copiées sur différents sites permettant de répartir les requêtes, soit parce qu'un site peut prendre la relève lorsque le serveur principal s'écroule. Une autre application tout aussi importante est la synchronisation des systèmes embarqués non connectés en permanence. [WEB_O6]

Chapitre 2: Les types de synchronisation

Ce qui peut se résumer à l'aide des trois types de scénarii suivants :

- deux serveurs distants sur lesquels les données doivent être consistantes ;
- deux serveurs, un comme serveur principal, l'autre comme serveur de backup à chaud ;
- plusieurs serveurs en cluster utilisés pour de l'équilibrage de charge et de la tolérance à la panne.

3. Principe

Le principe de la réplication, qui met en jeu au minimum deux SGBD, est assez simple et se déroule en trois temps :

1. La base maître reçoit un ordre de mise à jour (INSERT, UPDATE ou DELETE).
2. Les modifications faites sur les données sont détectées et stockées (dans une table, un fichier, une queue) en vue de leur propagation.
3. Un processus de réplication prend en charge la propagation des modifications à faire sur une seconde base dite esclave. Il peut bien entendu y avoir plus d'une base esclave.

Bien entendu il est tout à fait possible de faire de la réplication dans les deux sens (de l'esclave vers le maître et inversement). On parlera dans ce cas-là de réplication bidirectionnelle ou symétrique. Dans le cas contraire la réplication est unidirectionnelle (seulement du maître vers l'esclave) et on parle de réplication en lecture seule ou asymétrique. De plus la réplication peut être faite de manière synchrone ou asynchrone. Dans le premier cas la résolution des conflits éventuels entre deux sites interviennent avant la validation des transactions ; Dans le second cas, la résolution est faite dans des transactions séparées. Il est donc possible d'avoir quatre modèles de réplication : **[WEB_O6]**

- Réplication asymétrique (maître/esclave) avec propagation asynchrone ;
- Réplication asymétrique (maître/esclave) avec propagation synchrone ;
- Réplication symétrique ou Peer-to-peer (update everywhere) avec propagation asynchrone ;
- Réplication symétrique avec propagation synchrone.

4. La mise à jour synchrone

Dans la réplication synchrone, les données est déterminé à la base de données primaire ainsi que la réplication base de données en tant que partie de la même transaction. Ceci est également connu comme double commettre ou double phases. La transaction est écrite et s'engage à la fois le maître et l'esclave une partie de la transaction. Dans la réplication synchrone, la base de données primaire et répliquée tous

Chapitre 2: Les types de synchronisation

bases de données sont toujours en phase. **Figure 22** représente visuellement le processus de synchronisation réplication. [DOC_06]

La réplication synchrone est aussi appelée “réplication en temps réel”. La synchronisation est effectuée en temps réel puisque chaque requête est déployée sur l’ensemble des bases de données avant la validation (commit) de la requête sur le serveur où la requête est exécutée. Ce type de réplication assure un haut degré d’intégrité des données mais requiert une disponibilité permanente des serveurs et de la bande passante. Ce type de réplication, fortement dépendant des pannes du systèmes, nécessite de gérer des transactions multi sites coûteuses en ressources. [WEB_O6]

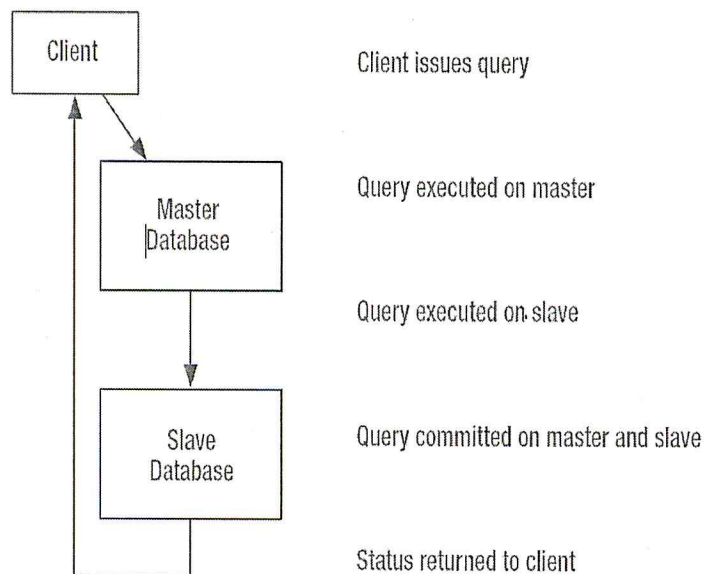


Figure 22 : La réplication synchrone

A requête est exécutée à la fois le maître et l'esclave, puis s'engage à la fois avant que le client reçoit le code de retour. Avec la requête la modification des données dans les deux endroits avant de retourner une réponse, toutes les bases de données dans l'environnement sont bien synchronisés.

4.1. La réplication synchrone asymétrique

Une modification sur le site primaire sera propagée aux sites secondaires à l'aide par exemple d'un trigger sur la table modifiée. La table est modifiée en temps réel sur les autres sites, ces modifications faisant parties de la transaction. [WEB_O6]

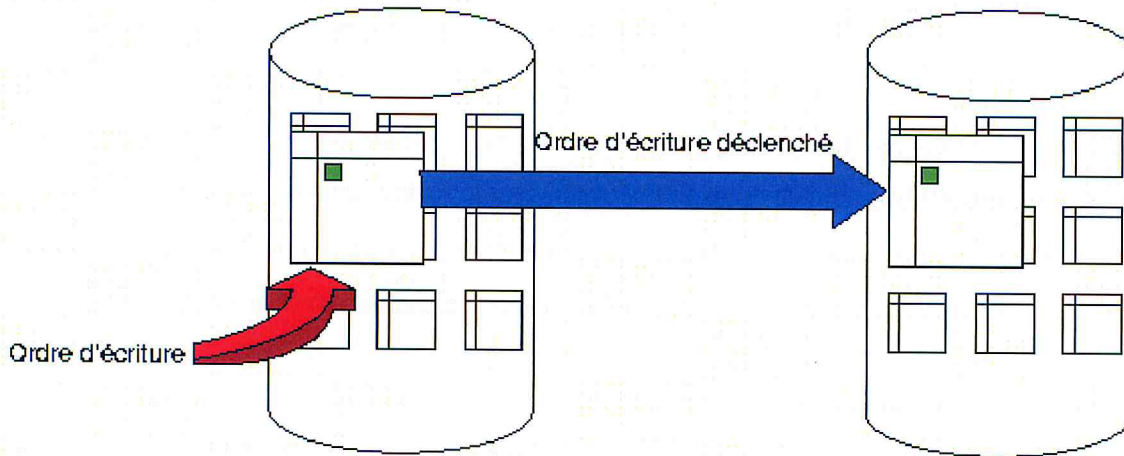


Figure 23 : La réplication synchrone asymétrique

4.2. La réplication synchrone symétrique

Dans ce cas, il n'y a pas de table maître, mais chaque table possède ses triggers, déclenchés lors d'une modification. Il est alors nécessaire de définir des priorités et de gérer les blocages des tables en attendant qu'une modification soit entièrement propagée. D'autre part, les triggers doivent différencier les mises à jour issues d'une réplication des mises à jour de requête directes. [WEB_O6]

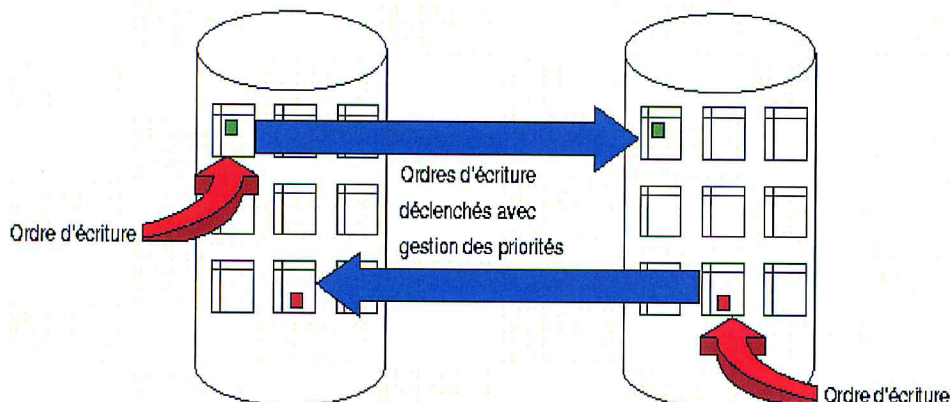


Figure 24 : La réplication synchrone symétrique

5. La mise à jour asynchrone

La réplication asynchrone signifie qu'une requête n'est pas ramassé sur les serveurs répliqués jusqu'à après la transaction est complète sur la base de données primaire. En règle générale, un processus tire ou pousse les modifications des données de la base de données primaire à un intervalle planifié et rend ces changements dans le système de réplication. En MySQL, les données sont extraites du maître par

5.1. La réplication asynchrone asymétrique

Les mises à jour sont stockées dans une file d'attente et ne seront propagées que lors d'un déclenchement programmé.

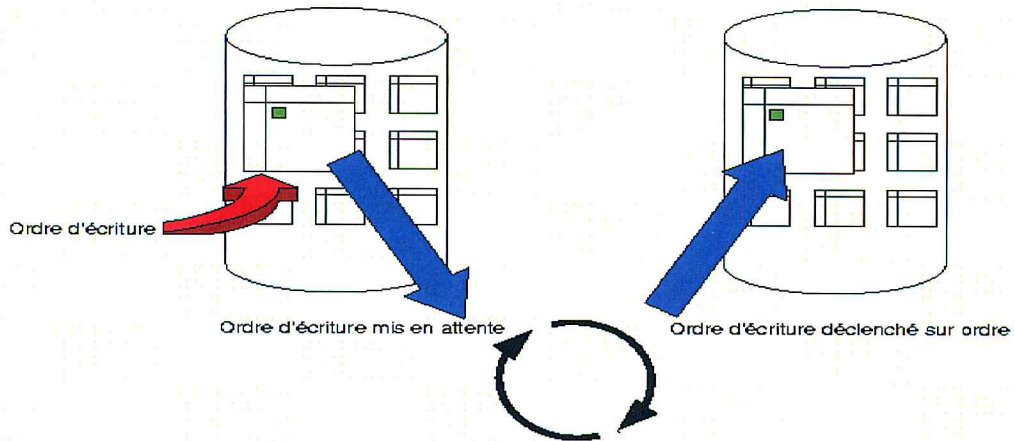


Figure 26 : La réplication asynchrone asymétrique

5.2. La réplication asynchrone symétrique

Toute modification sur toute table de toute base est stockée dans une file pour être rejouée ultérieurement. De fortes incohérences des données sont à craindre.

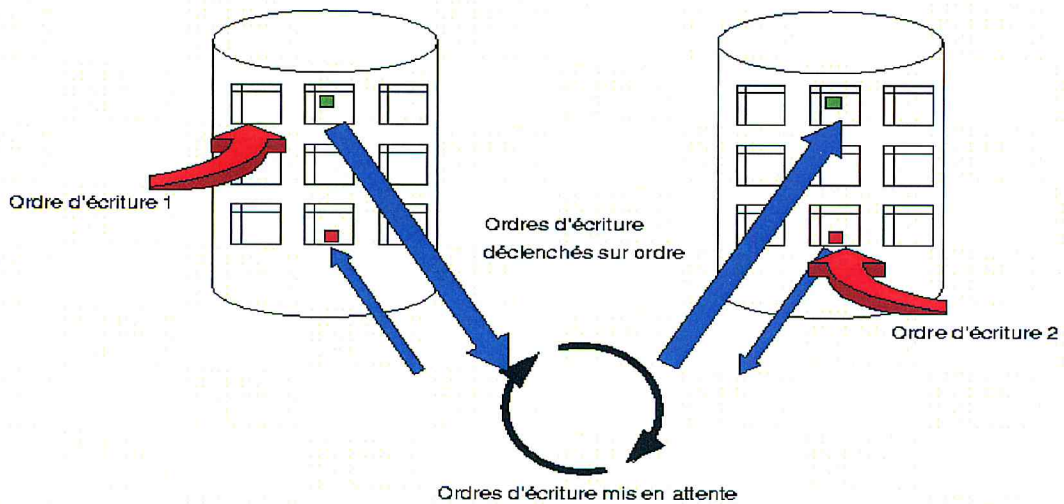


Figure 27: La réplication asynchrone symétrique

6 .Architecture de la synchronisation/réplication :

6.1. Maître de plusieurs esclaves

Le modèle de base de réplication plus, un seul maître par un ou plusieurs esclaves, Le maître est donné serveur ID 1 et chaque esclave dispose d'un ID différent. [DOC_06]

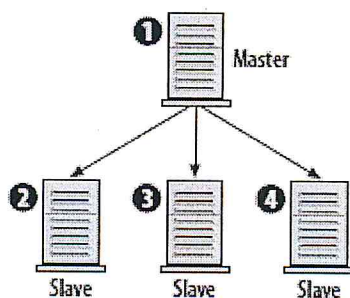


Figure 28 : maître de plusieurs esclaves

6.2. Esclave de deux maîtres

Ce serait bien d'utiliser un seul esclave pour gérer deux maîtres indépendants. Il permet de minimiser les coûts de matériel et ont toujours une sauvegarde serveur pour chaque maître. Cependant, c'est une violation de la deuxième règle: un esclave ne peut pas avoir deux maîtres. [DOC_06]

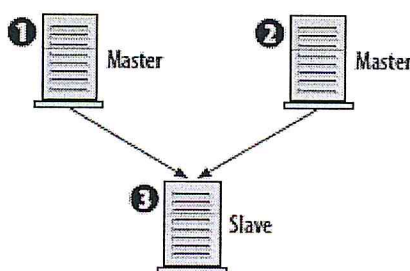


Figure 29 : Esclave de deux maîtres

6.3. Maître double

Une autre possibilité est d'avoir une paire de maîtres. Ceci est particulièrement utile lorsque deux parties séparées géographiquement d'une organisation ont besoin d'un accès en écriture à la même base de données partagée. Grâce à un design double-maître signifie que ni le site doit supporter la latence associée à une connexion WAN. [DOC_06]

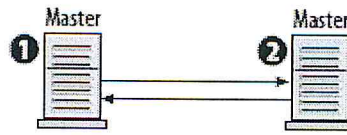


Figure 30 : Maître double

Il peut aussi

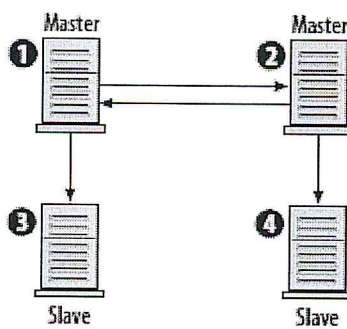


Figure 31: Maître double

6.4. Anneau de réplication (multi-maître)

La configuration double-maître est vraiment juste un cas particulier de la configuration en anneau de maître, il ya trois ou plusieurs maîtres qui former un cycle. Chaque serveur est un esclave de l'un de ses voisins et un maître à l'autre. [DOC_06]

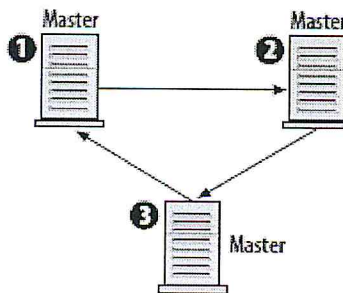


Figure 32: Anneau de réplication (multi-maître)

6.5. Pyramide

Dans les grandes organisations, diversité géographique, il peut y avoir un seul maître qui doit être répliquée à de nombreux petits bureaux. Plutôt que de configurer chaque esclave de contacter directement le maître, il peut être plus facile à gérer d'utiliser un symbole de la pyramide. [DOC_06]

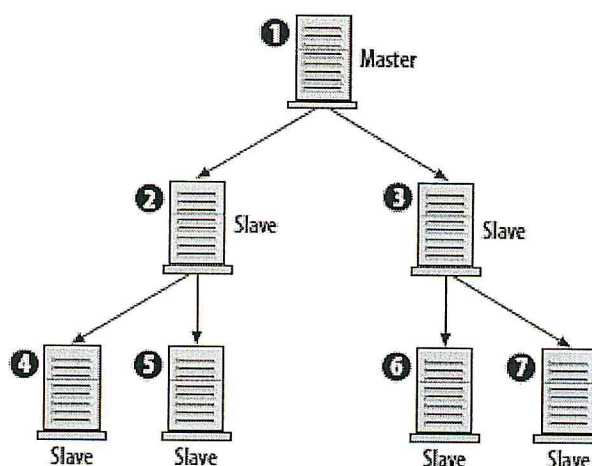


Figure 33: Pyramide

7. La synchronisation avec « pro sync framework »

7.1. Pourquoi Microsoft Sync Framework?

La capacité de charge de la synchronisation de ligne et hors ligne de données est la plus grande bénéficiant que ce cadre offre. L'objectif principal de Sync Framework est de permettre à n'importe quelle source de données à intégrer dans la synchronisation de données, quel que soit son type. Imaginez un scénario dans lequel les données de vente passés en mode déconnecté par un vendeur doit être en phase avec les données disponibles en ligne. Le vendeur peut-être l'aide d'un téléphone intelligent ou un assistant numérique personnel (PDA) pour se connecter au réseau d'entreprise à distance pour récupérer les dernières données et informations. Il est essentiel pour l'entreprise de s'assurer que les données et les informations sont cohérentes et en phase, quels que soient les modes dans lesquels ils sont examinés. C'est là que le Sync Framework entre en scène. [DOC_07]

Les états de documentation Sync Framework, "Microsoft Sync Services pour ADO.NET vous permet de synchroniser des données provenant de différentes sources sur deux niveaux, N-tier, et des architectures axées sur les services. Les services de synchronisation pour API ADO.NET pour le client et la synchronisation du serveur fournit un ensemble de composants pour synchroniser les données

Chapitre 2: Les types de synchronisation

entre les services de données et un magasin local, au lieu de seulement la réplication d'une base de données et son schéma ".

7.2. Principales caractéristiques du cadre de Microsoft Sync:

Un modèle de synchronisation puissant indépendant de la banque de données sous-jacente, des données Type ou le protocole utilisé , modèle de fournisseur extensible, Prise en charge des filtres et des données de résolution des conflits , Prise en charge de synchronisation des systèmes de fichiers, bases de données, et des extensions de partage simples (petites entreprises) telles que Really Simple Syndication (RSS) et flux Atome , Prise en charge peer-to-peer et topologies hub-and-spoke, Fonctionne avec n'importe quel nombre de points d'extrémité sur une configuration , Peut être utilisé de gestion ainsi que du code non managé. [DOC_07]

7.3. Flux de synchronisation

Fournisseurs de synchronisation permet de synchroniser les données entre les différents répliques. Répliques sont également connus comme les points d'extrémité ou les magasins de données. Les données réelles sont stockées dans le réplica. Nous avons besoin d'avoir un fournisseur de synchronisation pour chaque réplique de la réplique de synchroniser ses données avec d'autres répliques. Une réplique synchronise ses données avec une autre réplique en établissant une session de synchronisation.

Les fournisseurs de synchronisation communiquent entre eux en utilisant une session de synchronisation. Les deux prestataires de synchronisation sont fixés à l'agent de synchronisation, la synchronisation et la communication entre application initièrent les deux fournisseurs en utilisant l'agent de synchronisation. L'agent de synchronisation est responsable de l'établissement et la gestion de la session de synchronisation. Fournisseurs de synchronisation peuvent recevoir et appliquer les modifications aux répliques. Il existe deux fournisseurs de synchronisation sur le dessus du moteur d'exécution Sync Framework: le fournisseur source de synchronisation et de synchronisation destination fournisseur, respectivement.

Après avoir été invoqué par un agent de synchronisation, le fournisseur destination de synchronisation envoie ses connaissances au fournisseur source de synchronisation. Le fournisseur de source utilise cette connaissance pour déterminer les modifications et envoie sa connaissance de la destination. Le fournisseur de destination compare ses connaissances avec la source, résout les conflits, puis envoie la demande au fournisseur de source de données modifiées. Le fournisseur de source envoie les modifications apportées au fournisseur de destination, et le fournisseur de destination applique les modifications apportées à la réplique de destination. [DOC_07]

Chapitre 2: Les types de synchronisation

Dans une session de synchronisation, flux de synchronisation est toujours dans un seul sens. Ce que cela signifie est que le fournisseur de la source et le fournisseur de destination ne peuvent pas fonctionner simultanément. À n'importe quel moment donné dans une session de synchronisation, les flux d'informations entre la source et la destination des répliques, ou entre la source et la destination des répliques, mais ne s'écoule pas entre les deux simultanément. Dans sa forme simple, une session de synchronisation contient un agent de synchronisation et deux fournisseurs de synchronisation. L'un des fournisseurs est un fournisseur de source qui envoie les changements; l'autre est un fournisseur de destination qui reçoit et applique les modifications. Bien sûr, l'agent de synchronisation contrôle de ce flux.

Les fournisseurs de synchronisation illustrent un scénario dans lequel les métadonnées sont stockées dans le magasin de métadonnée intégrée offerte par Sync Framework, qui est très facile à utiliser et est construit au-dessus de la SQL Server CE. Vous pouvez également stocker les métadonnées dans votre propre boutique personnalisée. [DOC_07]

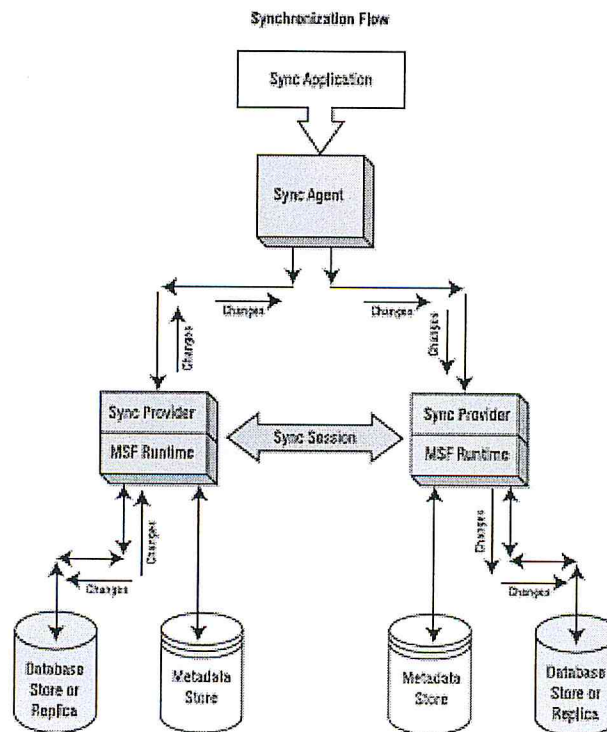


Figure 34: Flux de synchronisation

Chapitre 3

Conception des algorithmes de synchronisation

Chapitre 3 : Conception des algorithmes de synchronisation

1. Introduction

Nous avons vu dans le chapitre précédent que Joomla était composé de divers objets, les uns concernant le contenu (menus, articles, utilisateurs...) les autres ayant trait à l'apparence (templates) ou à des fonctionnalités supplémentaires (plug-ins).

Il est clair que la synchronisation pour nous ne signifie pas l'apparence ou les composants supplémentaire, celle-ci requérant une connaissance des différents objets ajoutés à la structure native de Joomla.

Par contre, et ce qui nous intéresse le plus, c'est de synchroniser le site en production et le site local où des interventions ont pu avoir lieu. Les interventions pouvant avoir lieu aussi bien sur le site local que sur le site distant, il est nécessaire –si on veut garder des sites miroirs du point de vue du contenu- de pouvoir synchroniser les quatre types d'objets formant le contenu : articles, catégories, utilisateurs, menus.

Dans ce chapitre nous allons présenter quatre algorithmes de synchronisation qui permettent de synchroniser les catégories, les articles, les utilisateurs et les menus.

2. Les types d'interventions et problèmes qu'ils engendrent pour la synchronisation

Les scénarios possibles sont nombreux. Toutefois, tous ne peuvent faire l'objet de synchronisation.

Imaginons, par exemple, qu'un même article a subi des modifications aussi bien sur le site local que sur le site distant. Quel serait le critère permettant de savoir s'il faut écraser la copie de l'article sur le site distant ou celle sur le site local. La date ? La priorité de l'éditeur ayant fait les modifications ? La fusion des deux versions de l'article est encore moins envisageable. Si l'un a supprimé une phrase, l'autre l'a remodelé, comment savoir laquelle garder ?

Ce genre de synchronisation ne va pas être pris en compte quitte à ce que les deux sites ne soient pas les miroirs l'un de l'autre sur ce point précis. Il appartiendra à l'administrateur de juger lequel gardé. Néanmoins, un outil pourrait permettre de repérer les articles ayant subi ce genre de traitements.

Chapitre 3 : Conception des algorithmes de synchronisation

Un autre type de synchronisation n'est pas plus envisageable. C'est le cas de la suppression. Si un article a été supprimé sur l'un des deux sites, comment savoir s'il faut le supprimer de l'autre site ou bien faut-il considérer que dans l'autre site, cet article est un nouvel article ?!

L'élaboration de critères fiables pour ce genre de synchronisation est difficile. Il devra peut-être prendre en compte le statut de l'éditeur ayant supprimé, la date de suppression... En tout état de cause, ce n'est pas chose facile.

Reste l'ajout dans l'un ou l'autre des deux sites ou les deux à la fois d'objets d'une des natures spécifiées précédemment.

Théoriquement, le schéma semble simple : ajouter tout Object x du site A n'appartenant pas à B au site B et/ou inversement.

L'opération serait simple mais les incidences sont à prendre en compte.

Imaginons que sur le site A un utilisateur s'est inscrit. Il aurait un numéro automatique. Si cet utilisateur est auteur, il y aurait des articles qui ont pour numéro d'auteur le numéro attribué par Joomla à cet auteur. Or si parallèlement, un autre auteur a été ajouté à B avec le même numéro, et que cet auteur a aussi écrit des articles, ces derniers auraient pour numéro d'auteur celui de B mais aussi celui du nouvel auteur de A. La synchronisation consistant à recopier « bêtement » pourrait avoir des conséquences graves.

Un autre problème –celui là d'autre technique- a surgit. Les concepteurs de Joomla ne donnent pas le schéma de leur base. Certains sites le proposent par reverse engineering. Il n'en demeure pas moins que lorsqu'on ajoute un article à la table `#_content`, toutes les entrées de la table `#_assets` changent. On ne comprend pas le pourquoi. Des membres du consortium Joomla ont bien voulu nous répondre nous disons que « c'était un secret ». Ils ne dévoilent pas le pourquoi.

Devant un problème aussi ardu, on ne peut qu'essayer de contourner l'obstacle.

Aussi avons-nous décelé deux types majeurs de synchronisations : ceux n'opérant pas d'opération secrète de la part de Joomla et ceux effectuant des changements radicaux.

Pour les seconds, il est impératif de supprimer les entrées dans la table cible et de recréer de nouvelles entrées à partir de la table source. Nous appellerons dans ce qui suit ce genre de synchronisation : synchronisation avec suppression. L'autre sera désigné synchronisation sans suppression.

Chapitre 3 : Conception des algorithmes de synchronisation

3. Les algorithmes de synchronisations avec suppression

3.1 Algorithme de synchronisation des catégories entre deux sites joomla

Comme on a vu dans le chapitre précédent, l'ajout ou la suppression d'une catégorie apporte des modifications sur deux tables de la base de données d'un site joomla, la table `#_assets` et la table `#_categories`.

La synchronisation des catégories entre la base de données du site local et la base de données du site distant nécessite de synchroniser ces deux tables.

On a deux sites A et B, un site en local et un autre site à distant

Si A est le site maître (celui qu'on considère comme source) alors l'algorithme consiste à supprimer toutes les données des tables `#_assets` et `#_categories` de la base de données du site B, récupérer toutes les données des tables `#_assets` et `#_categories` de la base de données du site A, et insérer toutes les données des tables `#_assets` et `#_categories` de la base de données du site A aux tables `#_assets` et `#_categories` de la base de données du site B.

Si B est le site maître alors l'opération est l'inverse de la précédente.

On peut résumer cet algorithme sur un schéma comme indiqué dans la **Figure 35**.

Chapitre 3 : Conception des algorithmes de synchronisation

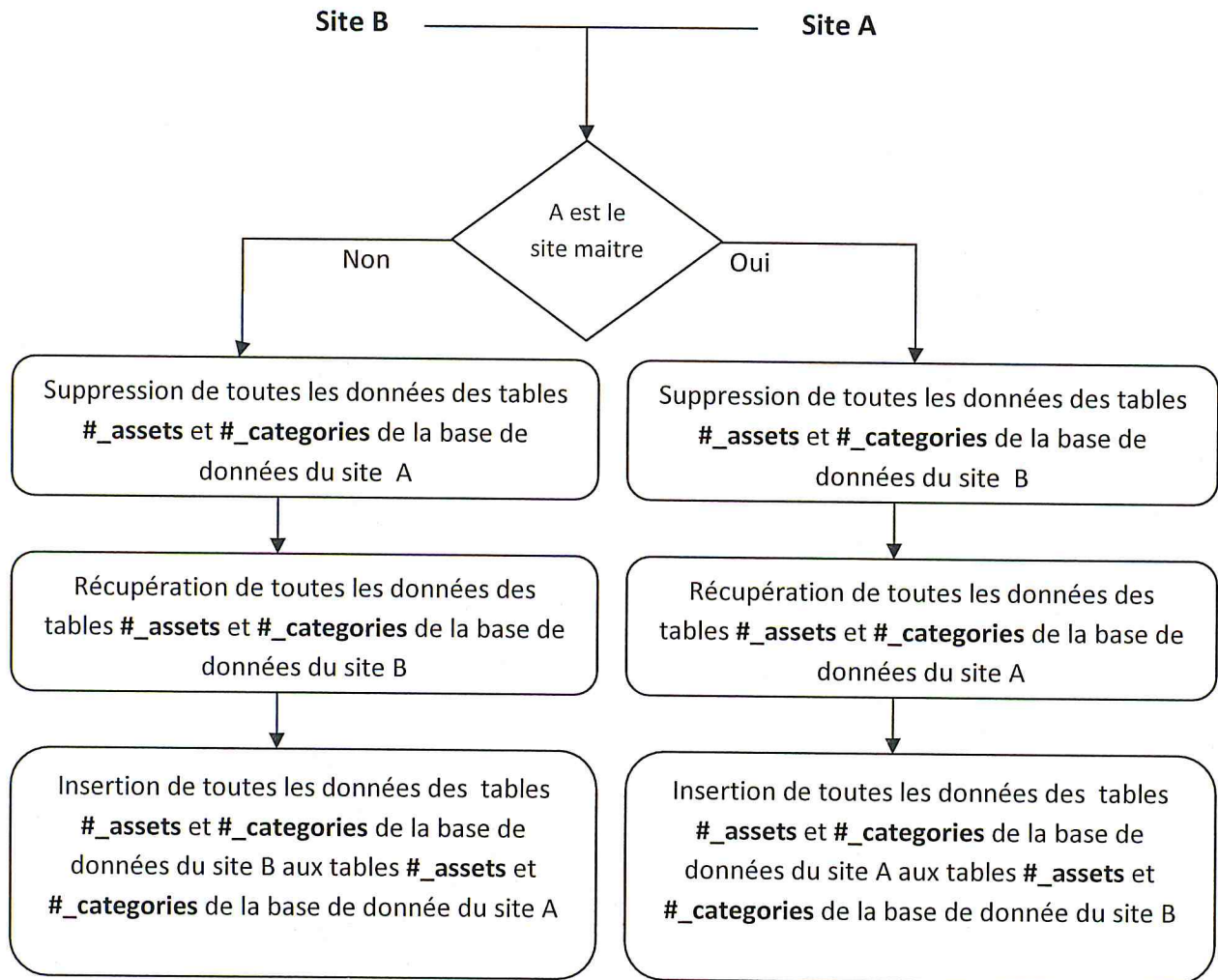


Figure 35: Algorithme de synchronisation des catégories avec suppression

Chapitre 3 : Conception des algorithmes de synchronisation

3.2 Algorithme de synchronisation des articles entre deux sites Joomla

Comme on a vu dans le chapitre précédent l'ajout ou la suppression d'un article apporte des modifications sur deux tables de la base de données d'un site Joomla la table **#_assets** et la table **#_content**.

La synchronisation des articles entre la base de données de site local et la base de données de site distant nécessite de synchroniser ces deux tables.

On a deux sites A et B, un site en local et un autre site à distant

Si A est le site maître (celui qu'on considère comme source) alors l'algorithme consiste à supprimer toutes les données des tables **#_assets** et **#_content** de la base de données du site B, récupérer toutes les données des tables **#_assets** et **#_content** de la base de données du site A, et insérer toutes les données des tables **#_assets** et **#_content** de la base de données du site A aux tables **#_assets** et **#_content** de la base de données du site B.

Si B est le site maître alors l'opération est l'inverse de la précédente.

On peut résumer cet algorithme sur un schéma comme indiqué dans la **Figure 36**.

Chapitre 3 : Conception des algorithmes de synchronisation

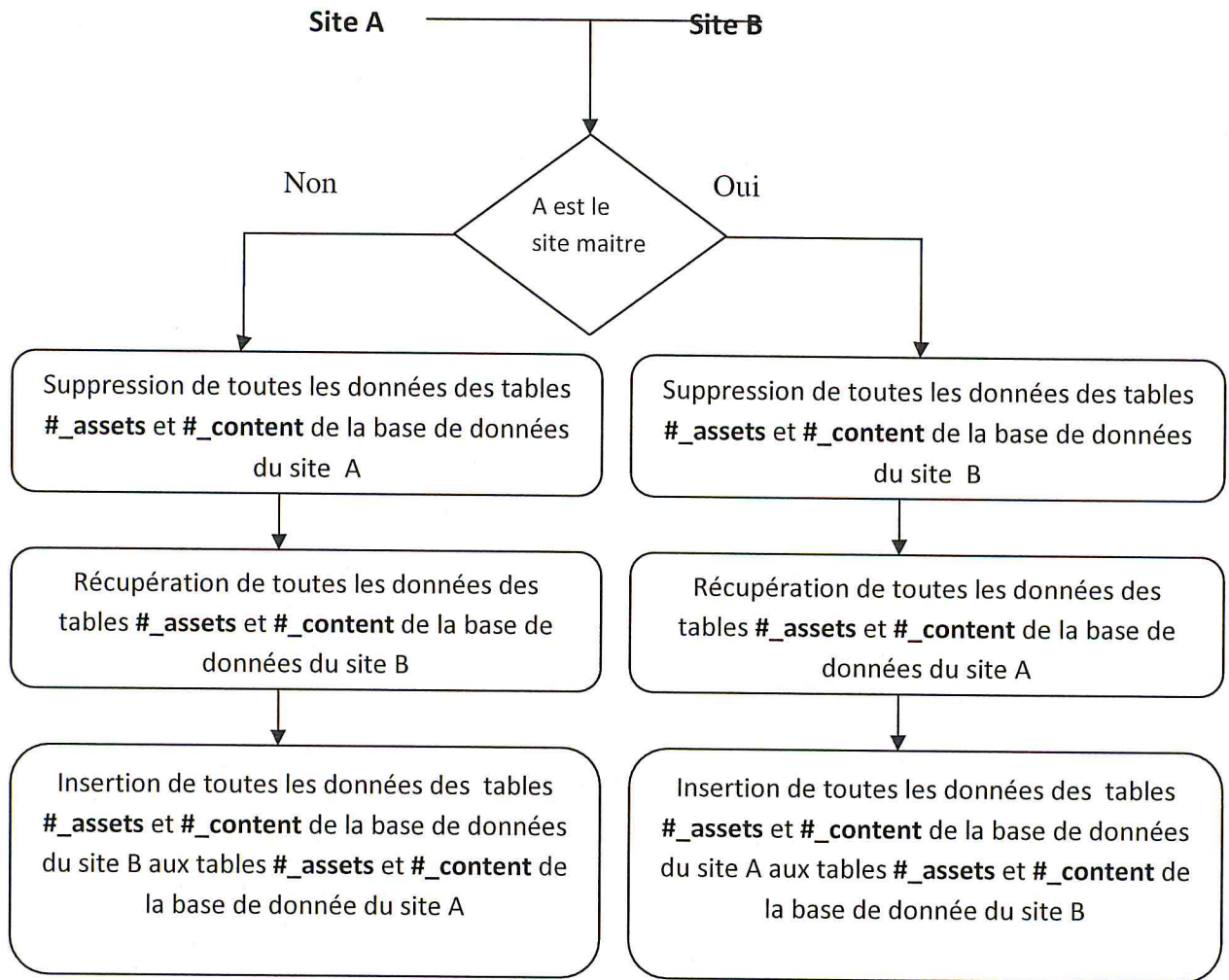


Figure 36: Algorithme de synchronisation des articles avec suppression

Chapitre 3 : Conception des algorithmes de synchronisation

3.3 Algorithme de synchronisation des utilisateurs entre deux sites Joomla

Comme on a vu dans le chapitre précédent l'ajout ou la suppression d'un utilisateur apporte des modifications sur deux tables de la base de données d'un site Joomla la table `#_users` et la table `#_user_usergroup_map`.

La synchronisation des utilisateurs entre la base de données de site local et la base de données de site distant nécessite de synchroniser ces deux tables.

On a deux sites A et B, un site en local et un autre site à distant

Si A est le site maître alors

Si A est le site maître (celui qu'on considère comme source) alors l'algorithme consiste à supprimer toutes les données des tables `#_users` et `#_user_usergroup_map` de la base de données du site B, récupérer toutes les données des tables `#_users` et `#_user_usergroup_map` de la base de données du site A, et insérer toutes les données des tables `#_users` et `#_user_usergroup_map` de la base de données du site A aux tables `#_users` et `#_user_usergroup_map` de la base de données du site B.

Si B est le site maître alors l'opération est l'inverse de la précédente.

On peut résumer cet algorithme sur un schéma comme indiqué dans la **Figure 37**.

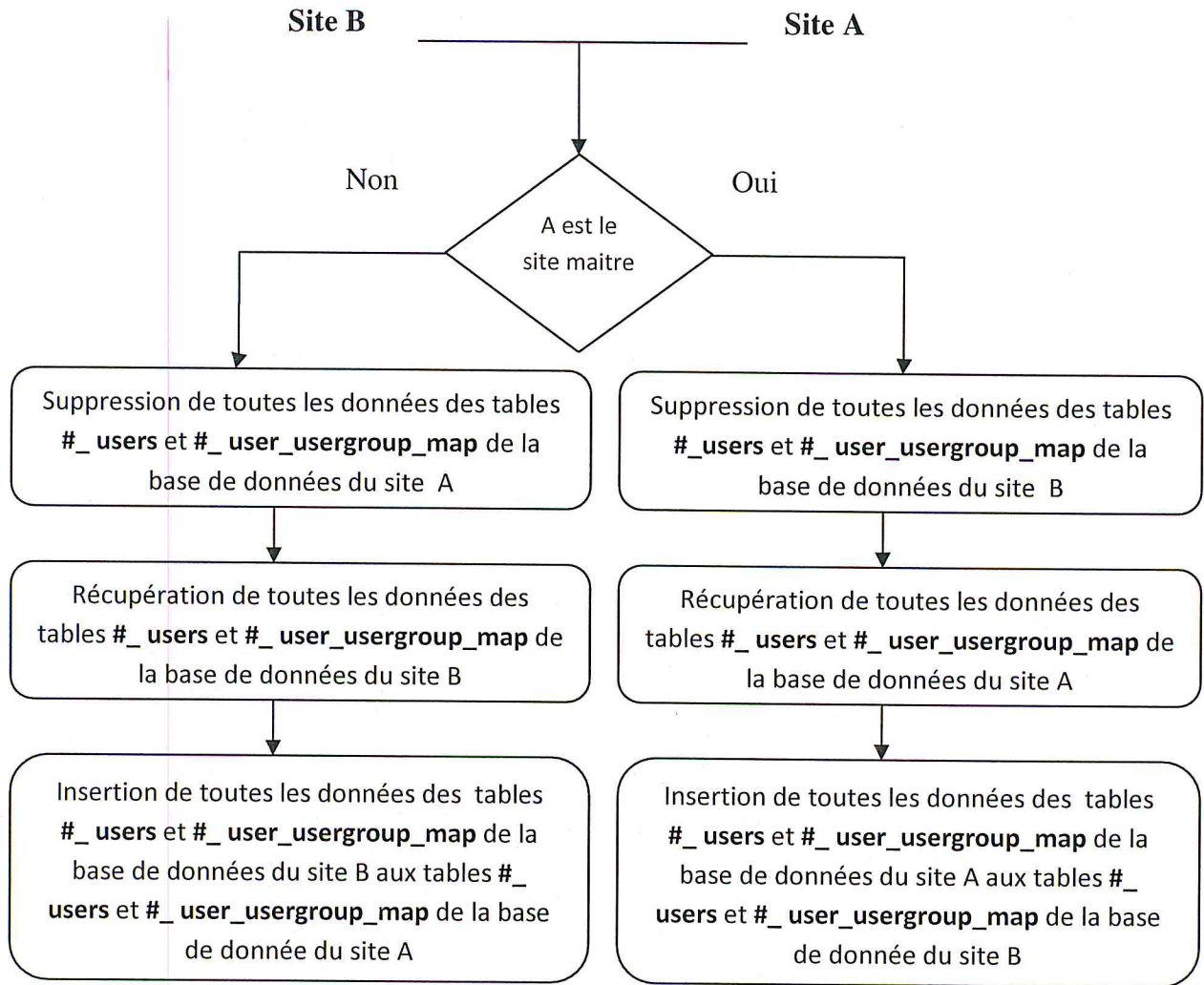


Figure 37: Algorithme de synchronisation des utilisateurs avec suppression

Chapitre 3 : Conception des algorithmes de synchronisation

3.4 Algorithme de synchronisation des menus entre deux sites Joomla

Comme on a vu dans le chapitre précédent l'ajout ou la suppression d'un article apporte des modifications sur deux tables de la base de données d'un site Joomla la table **#_menu_types**

La synchronisation des menus entre la base de données de site local et la base de données de site distant nécessite de synchroniser ces deux tables.

On a deux sites A et B, un site en local et un autre site à distant

Si A est le site maître alors

Si A est le site maître (celui qu'on considère comme source) alors l'algorithme consiste à supprimer toutes les données de la table **#_menu_types** de la base de données du site B, récupérer toutes les données de la table **#_menu_types** de la base de données du site A, et insérer toutes les données de la table **#_menu_types** de la base de données du site A au table **#_menu_types** de la base de données du site B.

Si B est le site maître alors l'opération est l'inverse de la précédente.

On peut résumer cet algorithme sur un schéma comme indiqué dans la **Figure 38**.

Chapitre 3 : Conception des algorithmes de synchronisation

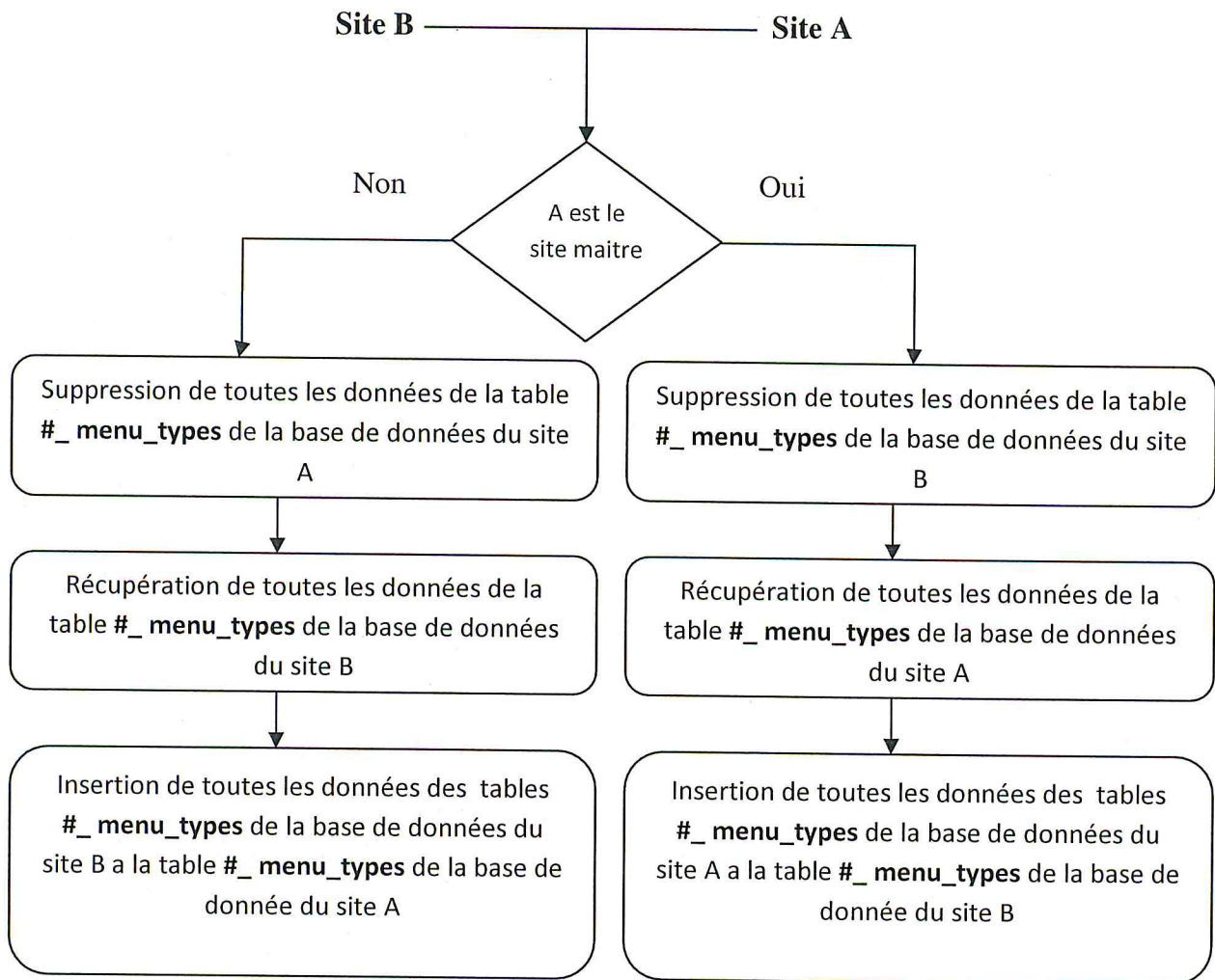


Figure 38: Algorithme de synchronisation des menus avec suppression

Chapitre 3 : Conception des algorithmes de synchronisation

4 .Algorithmes de synchronisations sans suppression

La logique de cette algorithmes consiste à créer deux tables dans la base de donnée du site maitre de joomla, une table qui contient des requêtes SQL (insérer, supprimer, modifier) et une deuxième table qui contient les paramètres des requêtes.

L'algorithmes permet de fouiller dans la table des requêtes de la base de données du site maitre, s'il a trouve des opérations il va les trie en suite il va transmettre les opérations SQL de la base de donnée du site maitre à la base de donnée du site esclave, après il va fouiller sur la table des paramètre du site maitre, s'il a trouver des paramètre il va les transmettre a la base de donnée du site esclave, En fin, l'algorithmes supprime toute les donnée des tables des requêtes et des paramètres pour éviter la duplication des données s'il y a une autre synchronisation.

Lors de l'ajout d'une catégorie on met sur la table des requêtes l'opération **insert into[#_assets]** et **insert into[#_categories]** et sur la table des paramètres on met les propriétés des catégories.

Lors de suppression d'une catégorie on met sur la table des requêtes l'opération **delete from [#_assets]** et **delete from [#_categories]** et sur la table des paramètres on met les propriétés des catégories.

Lors de l'ajout d'un article on met sur la table des requêtes l'opération **insert into[#_assets]** et **insert into[#_content]** et sur la table des paramètres on met les propriétés des articles.

Lors de suppression d'un article on met sur la table des requêtes l'opération **delete from [#_assets]** et **delete from [#_content]** et sur la table des paramètres on met les propriétés des articles.

Lors de l'ajout d'un utilisateur on met sur la table des requêtes l'opération **insert into[#_users]** et **insert into[#_user_usergroup_map]** et sur la table des paramètres on met les propriétés des utilisateurs.

Chapitre 3 : Conception des algorithmes de synchronisation

Lors de suppression d'un utilisateur on met sur la table des requêtes l'opération **delete from** [#_users] et **delete from** [#_user_usergroup_map] et sur la table des paramètres on met les propriétés des utilisateurs.

Lors de l'ajout d'un menu on met sur la table des requêtes l'opération **insert into** [#_menu_types] et sur la table des paramètres on met les propriétés des menus.

Lors de suppression d'un menu on met sur la table des requêtes l'opération **delete from** [#_menu_types] et sur la table des paramètres on met les propriétés des menus.

Ces opérations sont des étapes de préparation pour l'exécution de l'algorithme de synchronisation, Cet algorithme s'applique sur tous les cas de la synchronisation de la base de données des sites joomla (synchronisation des articles, les catégories, les menus et les utilisateurs).

On peut résumer cet algorithme sur un schéma comme indiquer dans la **Figure 39**.

Chapitre 3 : Conception des algorithmes de synchronisation

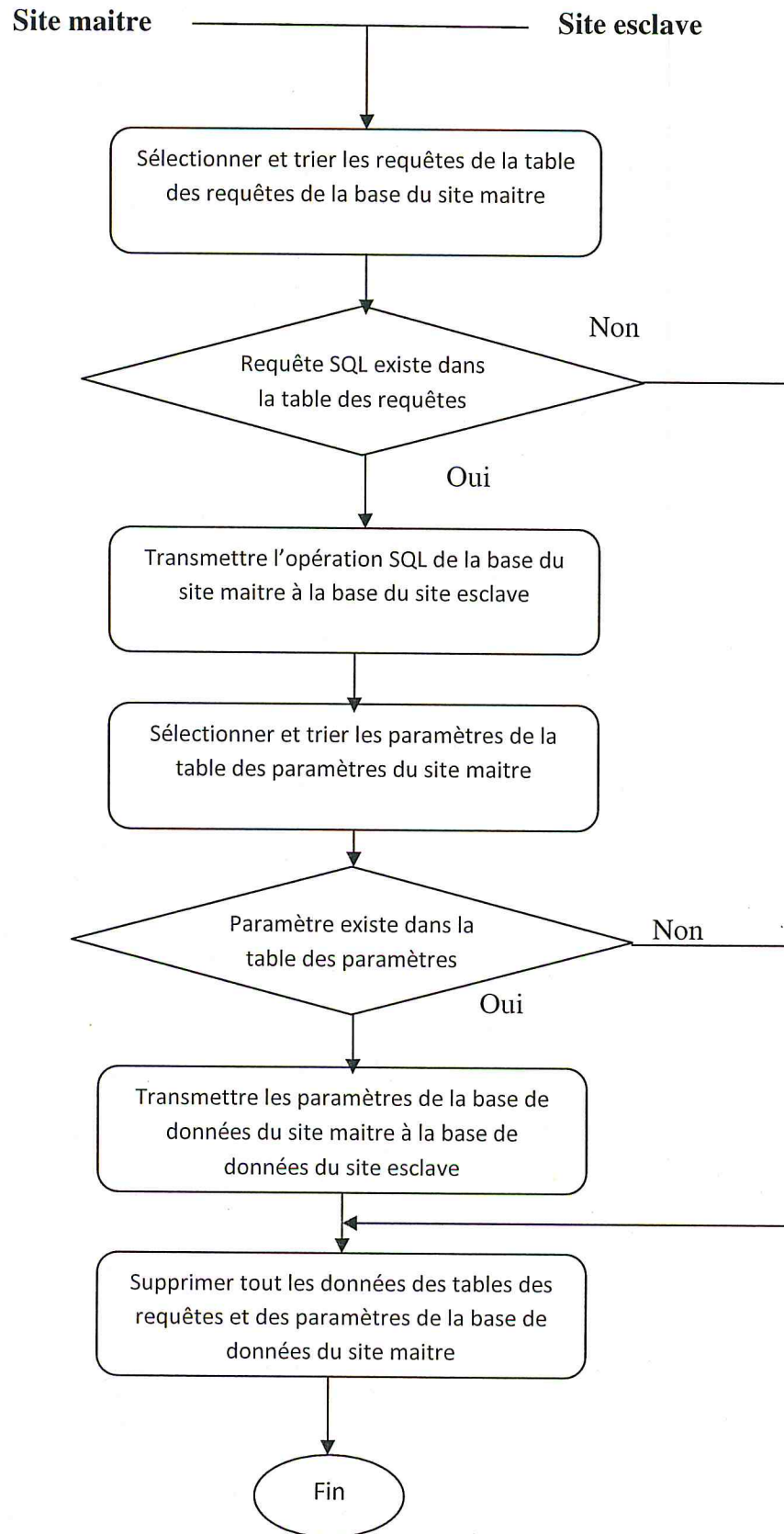


Figure 39: Algorithme de synchronisation sans suppression

Chapitre 3 : Conception des algorithmes de synchronisation

5. Conclusion

Dans ce chapitre nous avons détaillé les différents algorithmes de synchronisation des catégories, des articles, des utilisateurs et des menus. Nous avons expliqué chaque cas avec différents schémas.

Il reste que reste que nos algorithmes souffrent de ce qu'ils ne traitent pas les problèmes, ces derniers étant difficile à résoudre, ils ne font que les contourner.

Chapitre 4

Réalisation et test

Chapitre 4 : Réalisation et test

1-Introduction

Dans ce chapitre nous présenterons les principales parties d'une application basée sur le langage VB. Nous commencerons par l'architecture puis nous expliquons les outils d'implémentations et on dernier nous exposerons quelque pages de notre application.

2- Les outils d'implémentation

2.1. VB [WEB_O4]

Visual Basic est un environnement de développement du langage et de programmation créé par Microsoft. Il s'agit d'une extension du langage de programmation BASIC qui combine les fonctions de base et des commandes avec des contrôles visuels. Visual Basic fournit une interface utilisateur graphique GUI qui permet au développeur de faire glisser et déposer des objets dans le programme ainsi qu'écrire manuellement le code du programme.

2.2.Wampserver [WEB_O5]

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

2.3. Microsoft Access

Microsoft Access est un logiciel de base de données fonctionnant pour le système d'exploitation Microsoft Windows.

3. Synchronisation avec suppression

On a deux sites joomla un site en local et un site distant on veut synchroniser ces deux sites avec l'utilisation de notre outil de synchronisation.

Le site local est le site maitre et le site distant est le site esclave,

Le site local contient trois articles (Accueil, présentation et services), Un utilisateur (billel), Un menu (menu 1), Et une catégorie (categoriel), **Figure 40**

Chapitre 4 : Réalisation et test

Le site distant est un site de base qui contient que des fonctionnalités de base comme indiquer dans la **Figure 41**

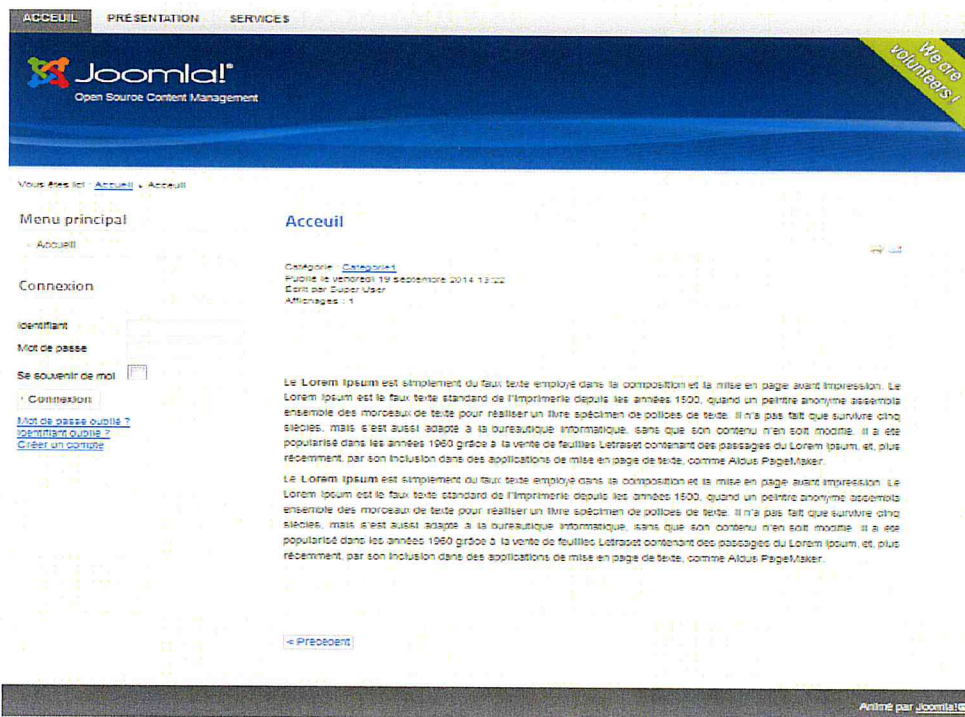


Figure 40: L'interface du site local

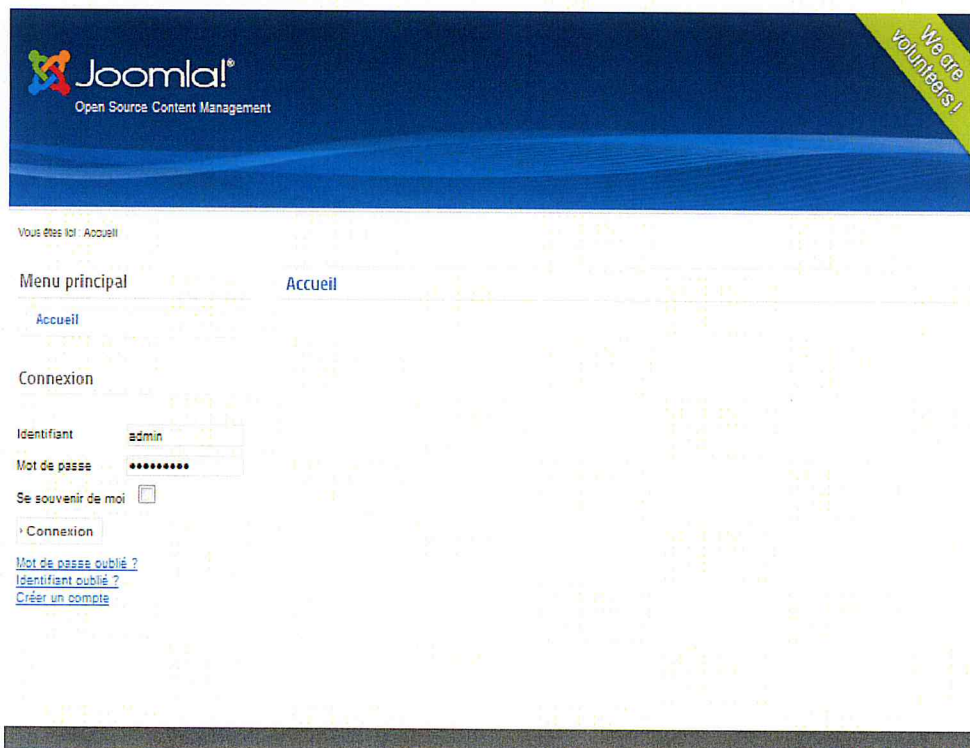


Figure 41: L'interface du site distant

Chapitre 4 : Réalisation et test

3.1. Impact sur la base de données du site local

Les articles (Accueil, présentation, services) appartiennent aux tables #_assets et #_content comme indiqué dans les Figure 42 et Figure 43

	id	asset_id	title	alias	title_alias	introtext	fulltext	state	sectionid	mask	catid
	23	1	59	60				1	com_templates		com_templates
	24	1	61	64				1	com_users		com_users
	25	1	65	68				1	com_weblinks		com_weblinks
	26	1	69	70				1	com_wrapper		com_wrapper
	27	8	18	19				2	com_content.category.2		Non catégorisé
	28	3	4	5				2	com_banners.category.3		Non catégorisé
	29	7	14	15				2	com_contact.category.4		Non catégorisé
	30	19	50	51				2	com_newsfeeds.category.5		Non catégorisé
	31	25	66	67				2	com_weblinks.category.6		Non catégorisé
	32	24	62	63				1	com_users.category.7		Non catégorisé
	33	1	71	72				1	com_finder		com_finder
	34	1	73	74				1	com_joomlaupdate		com_joomlaupdate
	35	8	20	27				2	com_content.category.8		Categorie1
	36	35	21	22				3	com_content.article.1		Accueil
	37	35	23	24				3	com_content.article.2		Présentation
	38	35	25	26				3	com_content.article.3		Services

Figure 42: La table #_assets après l'ajout des articles (accueil, presentation, services)

	id	asset_id	title	alias	title_alias	introtext	fulltext	state	sectionid	mask	catid
	1	36	Accueil	accueil		<p style="text-align: justify;"></p><p style="t...		1	0	0	8
	2	37	Présentation	presentation		<p style="text-align: justify;"></p><p style="t...		1	0	0	8
	3	38	Services	services		<p>Le Lorem Ipsum est simplement ...		1	0	0	8

Figure 43: La table #_content après l'ajout des articles (accueil, presentation, services)

La catégorie (categoriel) appartient aux tables #_assets et #_categories comme indiqué dans les figures Figure 44 et Figure 45

Chapitre 4 : Réalisation et test

	Afficher	Structure	SQL	Rechercher	Insérer	Exporter	Importer	Opérations	Déclencheurs	
<input type="checkbox"/>	Modifier	Copier	Effacer	23	1	59	60	1	com_templates	com_templates
<input type="checkbox"/>	Modifier	Copier	Effacer	24	1	61	64	1	com_users	com_users
<input type="checkbox"/>	Modifier	Copier	Effacer	25	1	65	68	1	com_weblinks	com_weblinks
<input type="checkbox"/>	Modifier	Copier	Effacer	26	1	69	70	1	com_wrapper	com_wrapper
<input type="checkbox"/>	Modifier	Copier	Effacer	27	8	18	19	2	com_content.category.2	Non catégorisé
<input type="checkbox"/>	Modifier	Copier	Effacer	28	3	4	5	2	com_banners.category.3	Non catégorisé
<input type="checkbox"/>	Modifier	Copier	Effacer	29	7	14	15	2	com_contact.category.4	Non catégorisé
<input type="checkbox"/>	Modifier	Copier	Effacer	30	19	50	51	2	com_newsfeeds.category.5	Non catégorisé
<input type="checkbox"/>	Modifier	Copier	Effacer	31	25	66	67	2	com_weblinks.category.6	Non catégorisé
<input type="checkbox"/>	Modifier	Copier	Effacer	32	24	62	63	1	com_users.category.7	Non catégorisé
<input type="checkbox"/>	Modifier	Copier	Effacer	33	1	71	72	1	com_finder	com_finder
<input type="checkbox"/>	Modifier	Copier	Effacer	34	1	73	74	1	com_joomlaupdate	com_joomlaupdate
<input type="checkbox"/>	Modifier	Copier	Effacer	35	8	20	27	2	com_content.category.8	Categorie1
<input type="checkbox"/>	Modifier	Copier	Effacer	36	35	21	22	3	com_content.article.1	Accueil
<input type="checkbox"/>	Modifier	Copier	Effacer	37	35	23	24	3	com_content.article.2	Présentation
<input type="checkbox"/>	Modifier	Copier	Effacer	38	35	25	26	3	com_content.article.3	Services

Figure 44: La table #_assets après l'ajout de la catégorie (Catégorie 1)

	Afficher	Structure	SQL	Rechercher	Insérer	Exporter	Importer	Opérations	Déclencheurs				
<input type="checkbox"/>	Modifier	Copier	Effacer	1	0	0	0	15	0	system	ROOT	root	
<input type="checkbox"/>	Modifier	Copier	Effacer	2	27	1	1	2	1	non-categorise	com_content	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	3	28	1	3	4	1	non-categorise	com_banners	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	4	29	1	5	6	1	non-categorise	com_contact	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	5	30	1	7	8	1	non-categorise	com_newsfeeds	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	6	31	1	9	10	1	non-categorise	com_weblinks	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	7	32	1	11	12	1	non-categorise	com_users	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	8	35	1	13	14	1	categorie1	com_content	Categorie1	categorie1

Figure 45: La table #_categories après l'ajout de la catégorie (Catégorie 1)

Le menu (menu1) est ajouté a la base de donnée au niveau de la table #_menu_types comme indiquer dans la figures Figure 46

Chapitre 4 : Réalisation et test

localhost » basesource » z4tpk_menu_types

Afficher Structure SQL Rechercher Insérer Exporter Importer Opé

✓ Affichage des lignes 0 - 1 (~2 total , Traitement en 0.0006 sec)

```
SELECT *
FROM `z4tpk_menu_types`
LIMIT 0, 30
```

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Trier sur l'index: Aucune

Options

	id	menutype	title	description
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	mainmenu	Menu principal	Le menu principal du site
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	menu1	Menu	Mon menu

Figure 46: La table #_menu_types après l'ajout d'un menu (menu1)

L'utilisateur bilbel est ajouté a la base de donnée au niveau des tables #_users et #_user_usergroup_map comme indiquer dans les Figure 47 et Figure 48

localhost » basesource » z4tpk_users

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Déclencheurs

```
SELECT *
FROM `z4tpk_users`
LIMIT 0, 30
```

Profilage [En ligne] [Modif]

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Trier sur l'index: Aucune

+ Options

	id	name	username	email	password	usertype	bl
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	97	Super User	admin	bilbelbouchoul@gmail.com	SP\$DEvYbTFY4mCAHK/cP37V31H		
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	98	bilbel	bilbelmaster	bilbel@gmail.com	SP\$D7R8wTNmf83uKQEOXQJcW73BD06Qx		

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 47: La table #_users après l'ajout d'un utilisateur (bilbel)

Chapitre 4 : Réalisation et test



Figure 48: La table #_user_usergroup_map après l'ajout d'un utilisateur (billel)

3.2. Impact sur la base de données du site distant avant la synchronisation

La base de donnée du site distant ne contient pas les articles (Accueil, présentation, services) au niveau des tables #_assets et #_content, comme indiquer dans les figures Figure 49 et Figure 50

<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	24	1	53	56	1 com_users com_users [{"core.admin":{"7":1},"core.manage":{"6":1},"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	25	1	57	60	1 com_weblinks com_weblinks [{"core.admin":{"7":1},"core.manage":{"6":1},"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	26	1	61	62	1 com_wrapper com_wrapper []
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	27	8	16	19	2 com_content category.2 Non catégorisé [{"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	28	3	4	5	2 com_banners category.3 Non catégorisé [{"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	29	7	14	15	2 com_contact category.4 Non catégorisé [{"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	30	19	42	43	2 com_newsfeeds category.5 Non catégorisé [{"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	31	25	56	59	2 com_weblinks category.6 Non catégorisé [{"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	32	24	54	55	1 com_users category.7 Non catégorisé [{"core.create":{"7":1},"core.delete":{"7":1},"core.edit":{"7":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	33	1	63	64	1 com_finder com_finder [{"core.admin":{"7":1},"core.manage":{"6":1}}]
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	34	1	65	66	1 com_joomlaupdate com_joomlaupdate [{"core.admin":{"7":1},"core.manage":{"6":1},"core.create":{"7":1},"core.delete":{"7":1}}]

Figure 49: La table #_assets du site distant avant la synchronisation des articles

Chapitre 4 : Réalisation et test

ID	Column Name	Type	Charset	Collation	Unsigned	Zero	Default Value	Actions
19	publish_up	datetime			Non	0000-00-00 00:00:00		Modifier, Supprimer
20	publish_down	datetime			Non	0000-00-00 00:00:00		Modifier, Supprimer
21	images	text	utf8_general_ci		Non	Aucune		Modifier, Supprimer
22	urls	text	utf8_general_ci		Non	Aucune		Modifier, Supprimer
23	attribs	varchar(5120)	utf8_general_ci		Non	Aucune		Modifier, Supprimer
24	version	int(10)			UNSIGNED	Non	1	Modifier, Supprimer
25	parentid	int(10)			UNSIGNED	Non	0	Modifier, Supprimer
26	ordering	int(11)				Non	0	Modifier, Supprimer
27	metakey	text	utf8_general_ci		Non	Aucune		Modifier, Supprimer
28	metadesc	text	utf8_general_ci		Non	Aucune		Modifier, Supprimer
29	access	int(10)			UNSIGNED	Non	0	Modifier, Supprimer
30	hits	int(10)			UNSIGNED	Non	0	Modifier, Supprimer
31	metadata	text	utf8_general_ci		Non	Aucune		Modifier, Supprimer
32	featured	tinyint(3)			UNSIGNED	Non	0	Modifier, Supprimer
33	language	char(7)	utf8_general_ci		Non	Aucune		Modifier, Supprimer
34	xreference	varchar(50)	utf8_general_ci		Non	Aucune		Modifier, Supprimer

Figure 50: La table #_content du site distant avant la synchronisation des articles

La base de donnée du site distant ne contient pas la catégorie (categoriel) au niveau des tables #_assets et #_categories, comme indiquer dans les figures Figure 51 et Figure 52

ID	Column Name	Type	Charset	Collation	Unsigned	Zero	Default Value	Actions
24	com_users	com_users					{ "core admin": {"7": 1}, "core manage": {"6": 1}, "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
25	com_weblinks	com_weblinks					{ "core admin": {"7": 1}, "core manage": {"6": 1}, "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
26	com_wrapper	com_wrapper					{ }	Modifier, Copier, Effacer
27	com_content.category.2	Non catégorisé					{ "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
28	com_banners.category.3	Non catégorisé					{ "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
29	com_contact.category.4	Non catégorisé					{ "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
30	com_newsfeeds.category.5	Non catégorisé					{ "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
31	com_weblinks.category.6	Non catégorisé					{ "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
32	com_users.category.7	Non catégorisé					{ "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
33	com_finder	com_finder					{ "core admin": {"7": 1}, "core manage": {"6": 1}, "core create": [], "core delete": [], "core edit": [] }	Modifier, Copier, Effacer
34	com_joomlaupdate	com_joomlaupdate					{ "core admin": [], "core manage": {"6": 1}, "core create": [], "core delete": [] }	Modifier, Copier, Effacer

Figure 51: La table #_assets du site distant avant la synchronisation des catégories

Chapitre 4 : Réalisation et test

localhost » basedestination » z4tpk_categories

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Déclencheurs

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

rier sur l'index: Aucune

Options

	id	asset_id	parent_id	lft	rgt	level	path	extension	title	alias	note	descripti
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	0	0	0	13	0		system	ROOT	root		
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	27	1	1	2	1	non-categorise	com_content	Non catégorisé	non-categorise		
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	28	1	3	4	1	non-categorise	com_banners	Non catégorisé	non-categorise		
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	29	1	5	6	1	non-categorise	com_contact	Non catégorisé	non-categorise		
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	5	30	1	7	8	1	non-categorise	com_newsfeeds	Non catégorisé	non-categorise		
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	6	31	1	9	10	1	non-categorise	com_weblinks	Non catégorisé	non-categorise		
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	7	32	1	11	12	1	non-categorise	com_users	Non catégorisé	non-categorise		

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 52: La table #_categories du site distant avant la synchronisation des catégories

La base de donnée du site distant ne contient pas l'utilisateur (billel) au niveau des tables #_users et #_user_usergroup_map, comme indiquer dans les figures Figure 53 et Figure 54

localhost » basedestination » z4tpk_users

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Dé

```
SELECT *  
FROM `z4tpk_users`  
LIMIT 0, 30
```

Profilage [En ligne] [Modifier]

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

+ Options

	id	name	username	email	password	usertype
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	188	Super User	admin	billelbouchoul@gmail.com	SP\$D3b2w.hjqe6/Vfse0SmstHnHdya2go.	deprece

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 53: La table #_users du site distant avant la synchronisation des utilisateurs

Chapitre 4 : Réalisation et test

localhost » basedestination » z4tpk_user_usergroup_map

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

Affichage des lignes 0 - 0 (~1 total), Traitement en 0.0005 sec

```
SELECT *
FROM `z4tpk_user_usergroup_map`
LIMIT 0, 30
```

Profilage [En ligne]

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

+ Options

	user_id	group_id
	188	8

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 54: La table #_user_usergroup_map du site distant avant la synchronisation des utilisateurs

La base de donnée du site distant ne contient pas le menu (menu) au niveau de table #_menu_types, comme indiquer dans la figure Figure 55

localhost » basedestination » z4tpk_menu_types

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

Affichage des lignes 0 - 0 (~1 total), Traitement en 0.0005 sec

```
SELECT *
FROM `z4tpk_menu_types`
LIMIT 0, 30
```

Profilage [En

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

- Options

	id	menutype	title	description
	1	mainmenu	Menu principal	Le menu principal du site

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Figure 55: La table #_menu_types du site distant avant la synchronisation des menus

3.3. Synchronisation

Pour la synchronisation revenons sur la l'interface de notre application

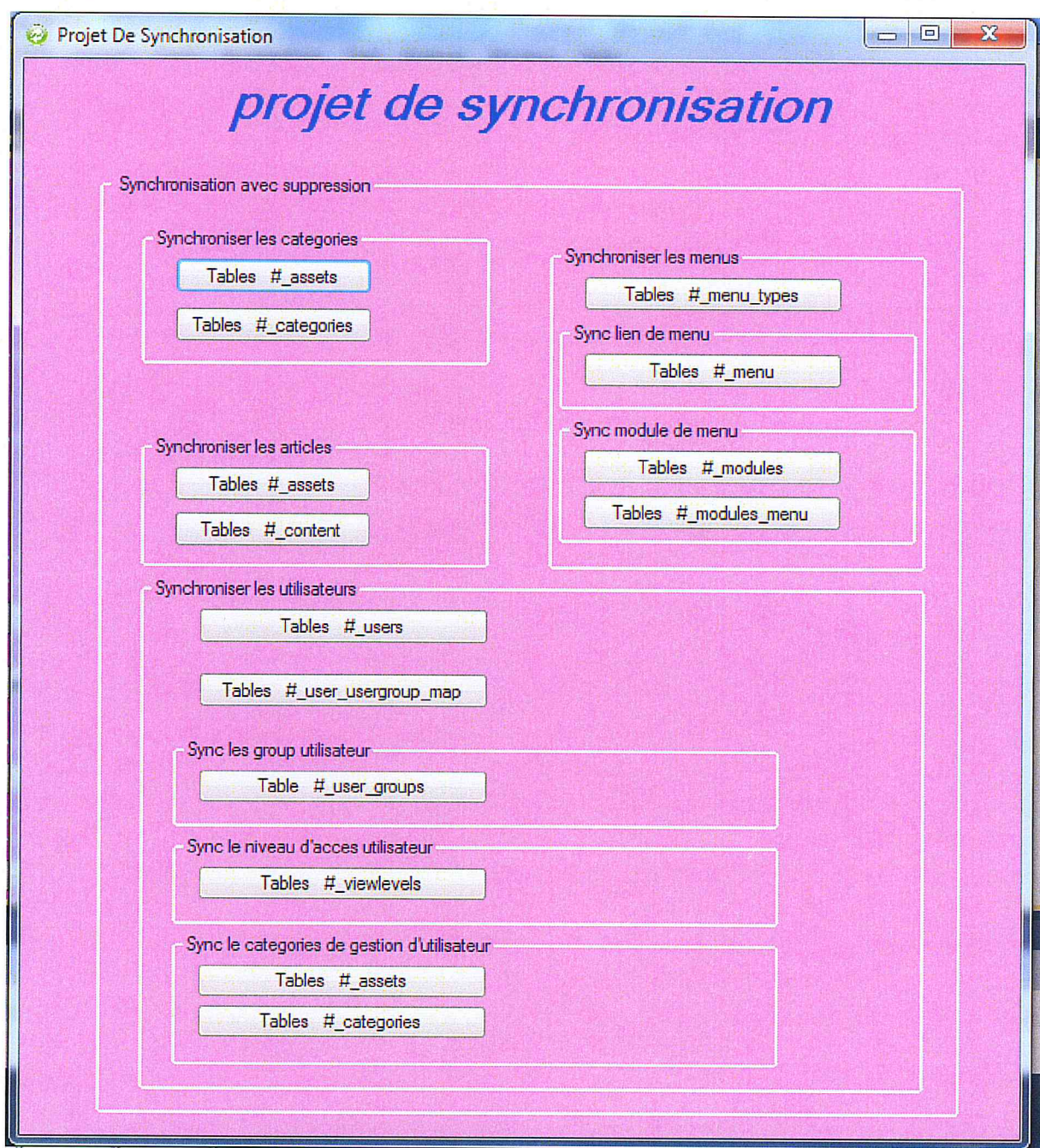


Figure 56: l'interface de l'application de synchronisation avec suppression

Chapitre 4 : Réalisation et test

Pour synchroniser les articles cliquant sur les boutons **Table #_assets** et **Table #_content**
Comme indiquer sur la **Figure 57**

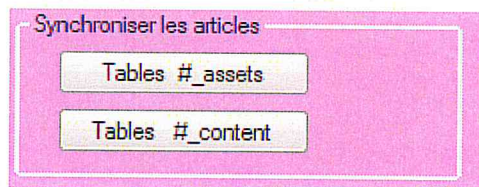


Figure 57: Les boutons de synchronisation des articles

Pour synchroniser les catégories cliquant sur les boutons **Table #_assets** et **Table #_catégories**, Comme indiquer sur la **Figure 58**



Figure 58: Les boutons de synchronisation des catégories

Pour synchroniser les utilisateurs cliquant sur les boutons **Table #_users** et **Table #_user_usergroup_map** Comme indiquer sur la figure **Figure 59**

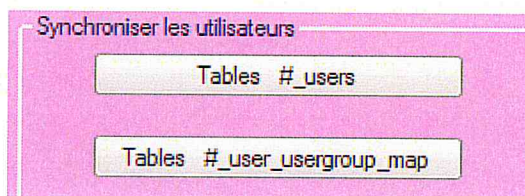


Figure 59: Les boutons de synchronisation des utilisateurs

Chapitre 4 : Réalisation et test

Pour synchroniser les menu cliquant sur les boutons **Table #_menu_types** Comme indiquer sur la figure **Figure 60**

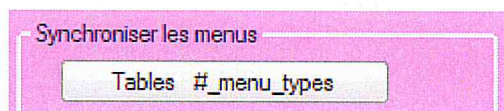


Figure 60: Les boutons de synchronisation des menus

3.4 .Impact sur la base de données du site distant après la synchronisation

Les articles (Accueil, présentation et services) qui en dans les tables #_assets et #_content de la base de donnée du site maitre sont recopies dans les tables #assets et #_content de la base de donnée du site esclave (distant),

Comme indiquer dans les figures **Figure 61** et **Figure 62**

A screenshot of a database management tool showing a table named '#_assets'. The table has 10 columns: ID, Name, Parent ID, Content, and several other columns. The rows represent different menu items, including 'com_newsfeeds.category.5', 'com_weblinks.category.6', 'com_users.category.7', 'com_finder', 'com_joomlaupdate', 'com_content.category.8', and three 'com_content.article' entries for 'Accueil', 'Présentation', and 'Services'. Each row includes icons for 'Modifier', 'Copier', and 'Effacer'.

ID	Name	Parent ID	Content	...
30	com_newsfeeds.category.5	19	50	51
31	com_weblinks.category.6	25	66	67
32	com_users.category.7	24	62	63
33	com_finder	1	71	72
34	com_joomlaupdate	1	73	74
35	com_content.category.8	8	20	27
36	com_content.article.1	35	21	22
37	com_content.article.2	35	23	24
38	com_content.article.3	35	25	26

Figure 61: La table #_assets du site distant après la synchronisation des articles

Chapitre 4 : Réalisation et test

	id	asset_id	title	alias	title_alias	introtxt	fulltext	state	sectionid	mask	catid	create
<input type="checkbox"/>	1	36	Accueil	accueil		<p style="text-align: justify;"> </p> <p style="t...		1	0	0	8	2014-0
<input type="checkbox"/>	2	37	Présentation	presentation		<p style="text-align: justify;"> </p> <p style="t...		1	0	0	8	2014-0
<input type="checkbox"/>	3	38	Services	services		<p>Le Lorem lpsum est simplement ...		1	0	0	8	2014-0

Figure 62: La table #_content du site distant après la synchronisation des articles

La catégorie (categoriel) qui es dans les tables #_assets et #_categories de la base de donnée du site maitre sont recopies dans les tables #assets et #_categories de la base de donnée du site esclave (distant),

Comme indiquer dans les figures **Figure 63** et **Figure 64**

	id	asset_id	title	alias	title_alias	introtxt	fulltext	state	sectionid	mask	catid	create
<input type="checkbox"/>	30	19	50	51	2	com_newsfeeds.category.5	Non catégorisé					
<input type="checkbox"/>	31	25	66	67	2	com_weblinks.category.6	Non catégorisé					
<input type="checkbox"/>	32	24	62	63	1	com_users.category.7	Non catégorisé					
<input type="checkbox"/>	33	1	71	72	1	com_finder	com_finder					
<input type="checkbox"/>	34	1	73	74	1	com_joomlaupdate	com_joomlaupdate					
<input type="checkbox"/>	35	8	20	27	2	com_content.category.8	Categorie1					
<input type="checkbox"/>	36	35	21	22	3	com_content.article.1	Accueil					
<input type="checkbox"/>	37	35	23	24	3	com_content.article.2	Présentation					
<input type="checkbox"/>	38	35	25	26	3	com_content.article.3	Services					

Figure 63: La table #_assets du site distant après la synchronisation des catégories

Chapitre 4 : Réalisation et test

localhost » basedestination » z4tpk_categories													
Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Déclencher													
												#_assets table.	
<input type="checkbox"/>	Modifier	Copier	Effacer	1	0	0	0	15	0		system	ROOT	root
<input type="checkbox"/>	Modifier	Copier	Effacer	2	27	1	1	2	1	non-categorise	com_content	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	3	28	1	3	4	1	non-categorise	com_banners	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	4	29	1	5	6	1	non-categorise	com_contact	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	5	30	1	7	8	1	non-categorise	com_newsfeeds	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	6	31	1	9	10	1	non-categorise	com_weblinks	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	7	32	1	11	12	1	non-categorise	com_users	Non catégorisé	non-categorise
<input type="checkbox"/>	Modifier	Copier	Effacer	8	35	1	13	14	1	categorie1	com_content	Categorie1	categorie1

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 64: La table #_categories du site distant après la synchronisation des catégories

L'utilisateur **billem** qui es dans les tables #_users et #_user_usergroup_map de la base de donnée du site maitre sont recopies dans les tables #users et #_user_usergroup_map de la base de donnée du site esclave (distant),

Comme indiquer dans les figures **Figure 65** et **Figure 66**

localhost » basedestination » z4tpk_users												
Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Dé												
<input type="checkbox"/> Profilage [En ligne] [Modifier] [
Afficher : Ligne de départ: <input type="text" value="0"/> Nombre de lignes: <input type="text" value="30"/> En-têtes à intervalle de <input type="text" value="100"/> lignes												
Trier sur l'index: <input type="text" value="Aucune"/>												
+ Options												
← T →												
				id	name	username	email				pssword	
<input type="checkbox"/>	Modifier	Copier	Effacer	97	Super User	admin	billem@bouchoul@gmail.com				\$P\$DEVYbTFY4mCAHK/cP37V31HAWAERmd1	
<input type="checkbox"/>	Modifier	Copier	Effacer	98	billem	billemmaster	billem@gmail.com				\$P\$D7R8wTNmf83uKQEOXQEJcW73BD06Qx	

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 65: La table #_users du site distant après la synchronisation des utilisateurs

Chapitre 4 : Réalisation et test

localhost » basedestination » z4tpk_user_usergroup_map

Afficher Structure SQL Rechercher Insérer Exporter Imp

```
SELECT *  
FROM `z4tpk_user_usergroup_map`  
LIMIT 0, 30
```

Profilage [En ligne]

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100

Trier sur l'index: Aucune

+ Options

	user_id	group_id
	Foreign Key to #_users.id	Foreign Key to #_usergroups.id
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	97	8
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	98	2

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 66: La table #_user_usergroup_map du site distant après la synchronisation des utilisateurs

Le menu (menu1) qui es dans la table #_menu_type de la base de donnée du site maitre sont recopies dans les tables #_menu_types de la base de donnée du site esclave (distant),

Comme indiquer dans la figure Figure 67

localhost » basedestination » z4tpk_menu_types

Afficher Structure SQL Rechercher Insérer Exporter Importer Op

```
SELECT *  
FROM `z4tpk_menu_types`  
LIMIT 0, 30
```

Profilage [En ligne] [Modifier] [Ex

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Trier sur l'index: Aucune

+ Options

	id	menutype	title	description
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	mainmenu	Menu principal	Le menu principal du site
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	menu1	Menu	Mon menu

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Figure 67: La table #_menu_types du site distant après la synchronisation des menus

Chapitre 4 : Réalisation et test

3.5 .Impact sur le site distant après la synchronisation

Après la synchronisation des bases de données, le site distant devient identique au site local.

Comme indiqué dans la figure **Figure 68**

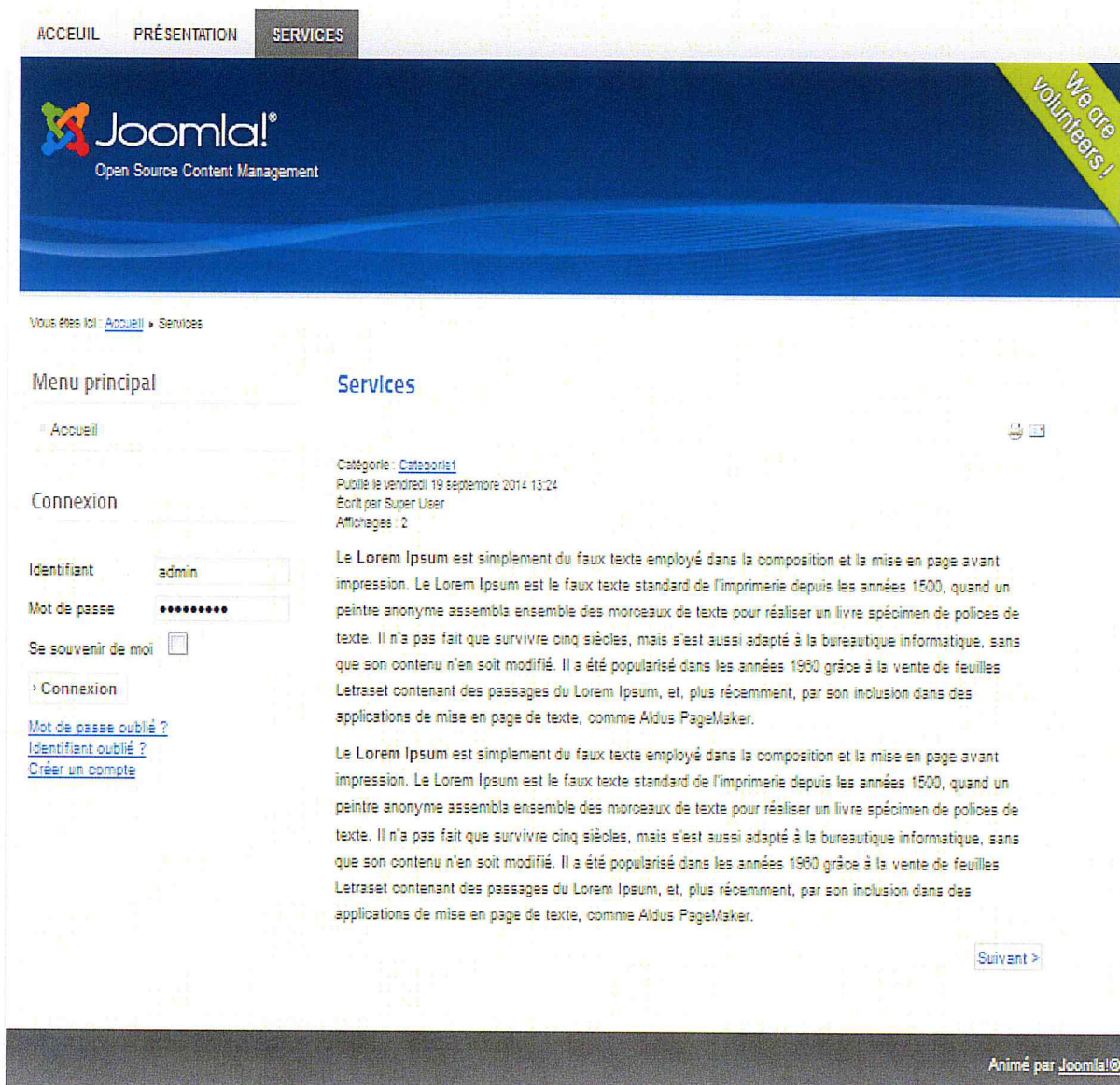


Figure 68: Le site distant après la synchronisation

Chapitre 4 : Réalisation et test

4 .Synchronisation sans suppression

Le principe de synchronisation de cet algorithme nécessite la création de deux tables, un table pour les requêtes SQL et un autre table pour les paramètres des requêtes SQL.

On a deux sites joomla un site en local et un site distant on veut synchroniser ces deux sites avec l'utilisation de notre outil de synchronisation sans suppression des donnée du site esclave, Le site local est le site maitre et le site distant est le site esclave.

4.1. Synchronisation des utilisateurs

4.1.1. Cas d'ajout d'un utilisateur

On ajoute un utilisateur (billel) qui va enregistrer sur les tables `#_users` et `#_user_usergroup_map` de la base de donnée du site maitre (local).

Comme indiquer dans les figures **Figure 69** et **Figure 70**

The screenshot shows a database management interface for the table `#_users`. The table contains the following data:

	id	name	username	email	psd	usertype	bl
<input type="checkbox"/>	941	Super User	admin	billelbouchoul@gmail.com	\$PSDEzJKO5KnlugfTQ7m.e.mMMtDrcHdT0	deprecated	
<input type="checkbox"/>	942	billel	bob	billel@gmail.com	\$PSDID568HEgblBKotHAdD8Z3sqU0kNTF0		

Figure 69:La table `#_users` du site maitre après l'ajout d'un utilisateur **billel**

Chapitre 4 : Réalisation et test

localhost » 12 » a_user_usergroup_map

Afficher Structure SQL Rechercher Insérer Exporter Importer

```
SELECT *
FROM #_user_usergroup_map
LIMIT 0, 30
```

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lig

rier sur l'index: Aucune

Options

	user_id	group_id
	Foreign Key to #_users.id	Foreign Key to #_usergroups.id
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	941	8
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	942	2

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 70: La table #_user_usergroup_map du site maitre après l'ajout d'un utilisateur **billel**

Cette utilisateur n'existe pas dans les tables #_users et #_user_usergroup_map de la base de donnée du site distant

Comme indiquer dans les figures Figure 71 et Figure 72

localhost » 13 » a_users

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations D

```
SELECT *
FROM #_users
LIMIT 0, 30
```

Profilage [En ligne] [Modifier] [Explique

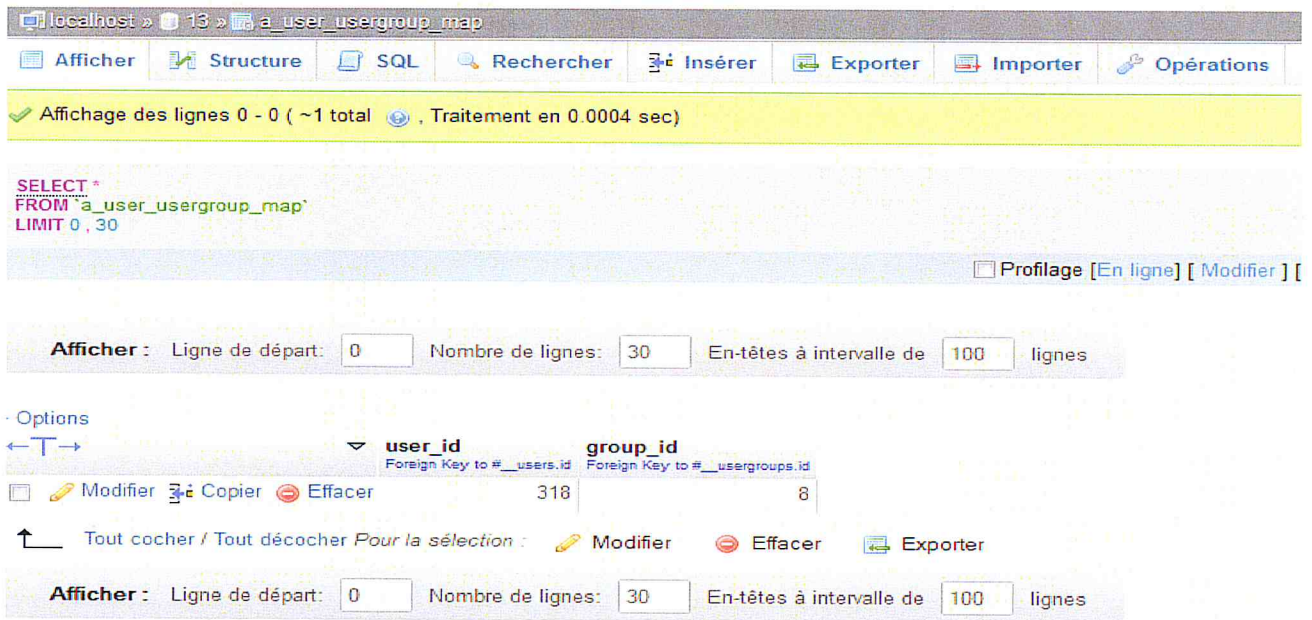
Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Options

	id	name	username	email	psd	user
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	318	Super User	admin	billelbouchoul@gmail.com	SP\$DX30t3JImZDOIkLZu7LTGoUM4HmZTE1	dep

Figure 71: La table #_users du site esclave avant la synchronisation

Chapitre 4 : Réalisation et test



localhost » 13 » #_user_usergroup_map

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

Affichage des lignes 0 - 0 (~1 total) Traitement en 0.0004 sec

```
SELECT *  
FROM `#_user_usergroup_map`  
LIMIT 0, 30
```

Profilage [En ligne] [Modifier]

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Options

	user_id	group_id
	Foreign Key to #__users.id	Foreign Key to #__usergroups.id
Modifier Copier Effacer	318	8

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Figure 72: La table #_users du site esclave avant la synchronisation

Pour synchroniser les utilisateurs, On va ajouter les requêtes suivantes à la table des requêtes:

insert into

```
[#_users](id,name,username,email,password,usertype,block,sendEmail,registerDate,lastvisitDate,activation,params,lastResetTime,resetCount) values (:0,:1,:2,:3,:4,:5,:6,:7,:8,:9,:10,:11,:12,:13)
```

```
Insert into [#_user_usergroup_map](user_id,group_id) values (:0,:1)
```

Comme indiquer dans la figure **Figure 73**

En suite, on ajoute les paramètres de l'utilisateur qu'on a ajouté sur la table des paramètres, comme indiquer dans la figure **Figure 74**

Chapitre 4 : Réalisation et test

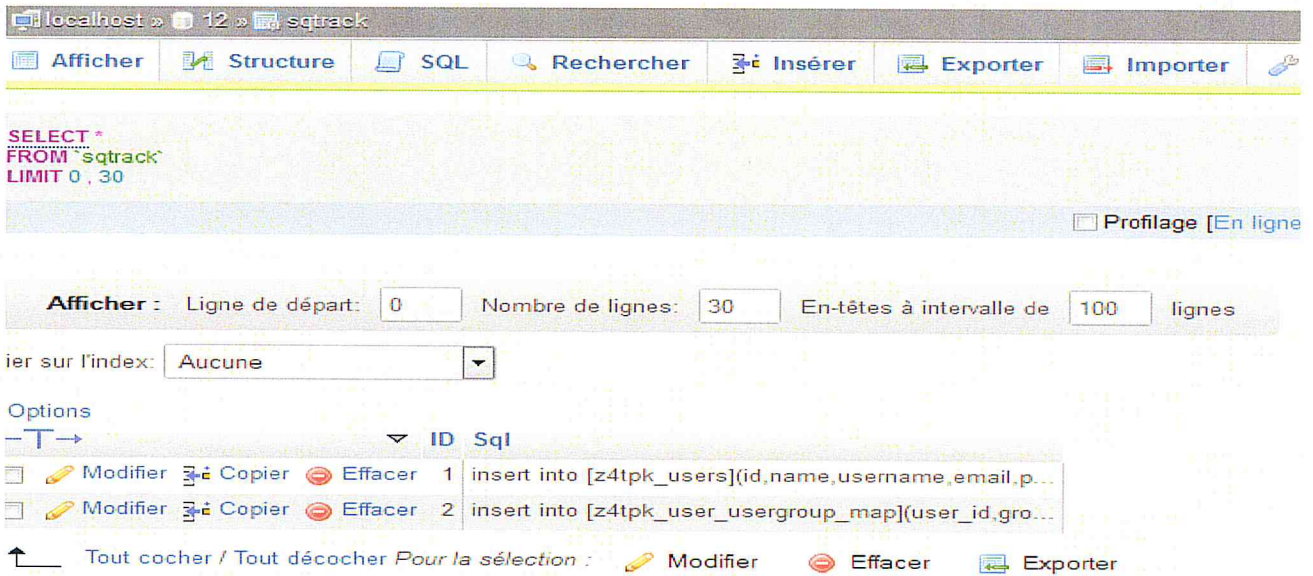


Figure 73:La table des requêtes SQL des utilisateurs

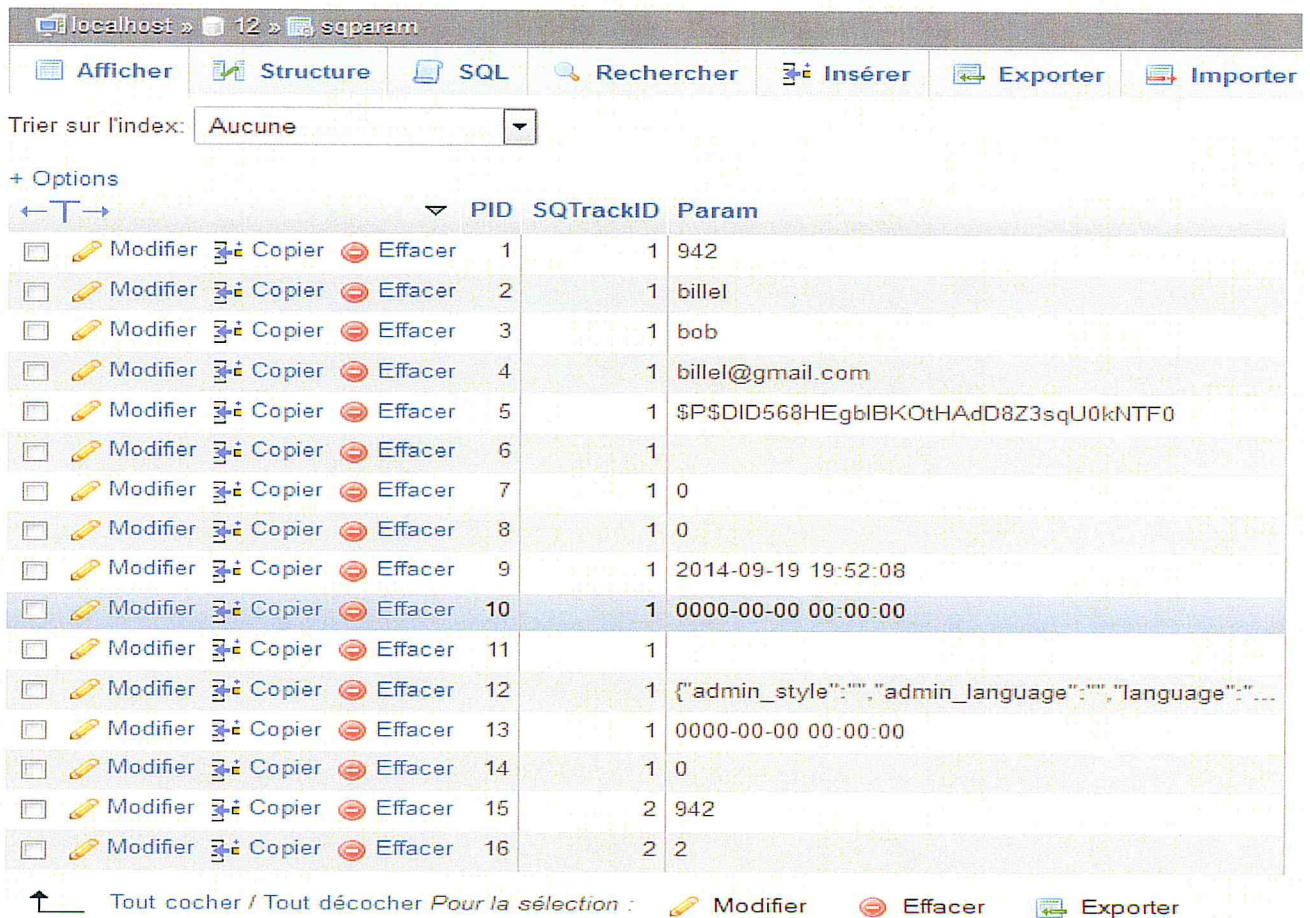


Figure 74:La table des paramètres des utilisateurs

Chapitre 4 : Réalisation et test

Et on exécute l'algorithme de synchronisation

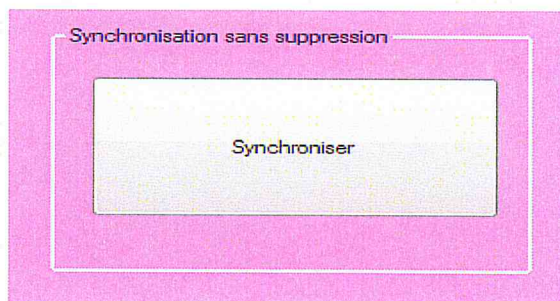
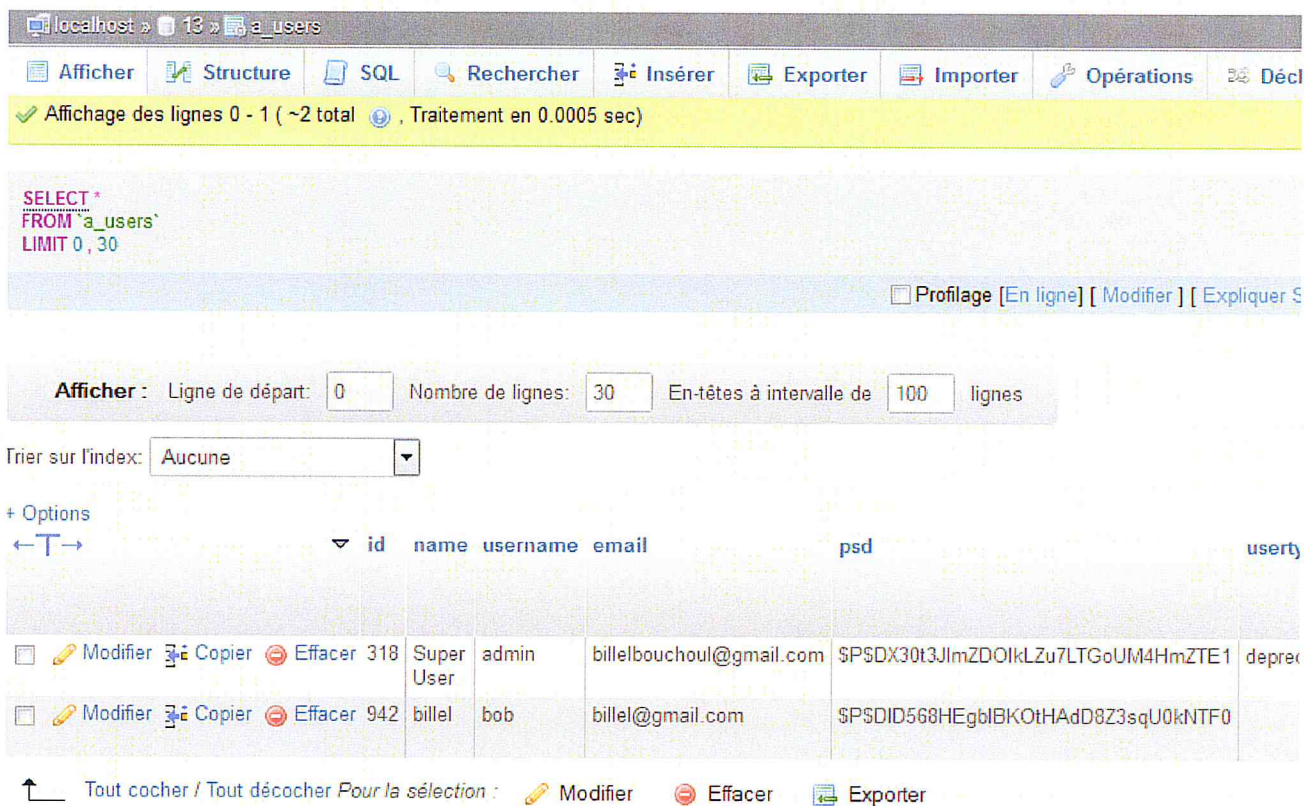


Figure 75: Bouton de synchronisation sans suppression

L'utilisateur va enregistrer sur les tables `#_users` et `#_user_usergroup_map` de la base de donnée du site esclave, comme indiquer dans les figures **Figure 76** et **Figure 77**



The screenshot shows a database management interface for a MySQL database. The top bar indicates the connection to 'localhost' at port '13' for the database 'a_users'. The main area displays the SQL query: `SELECT * FROM `a_users` LIMIT 0, 30`. Below the query, there are controls for displaying rows: 'Afficher : Ligne de départ: 0', 'Nombre de lignes: 30', and 'En-têtes à intervalle de 100 lignes'. The table data is as follows:

	id	name	username	email	psd	userty
<input type="checkbox"/>	318	Super User	admin	billemouchoul@gmail.com	SP\$DX30t3JImZD0IkLZu7LTGoUM4HmZTE1	deprex
<input type="checkbox"/>	942	billem	bob	billem@gmail.com	SP\$DID568HEgblBK0tHAdD8Z3sqU0kNTF0	

At the bottom, there are options to 'Tout cocher / Tout décocher' and actions for 'Modification', 'Suppression', and 'Exportation'.

Figure 76: La table `#_users` du site distant après la synchronisation sans suppression

Chapitre 4 : Réalisation et test

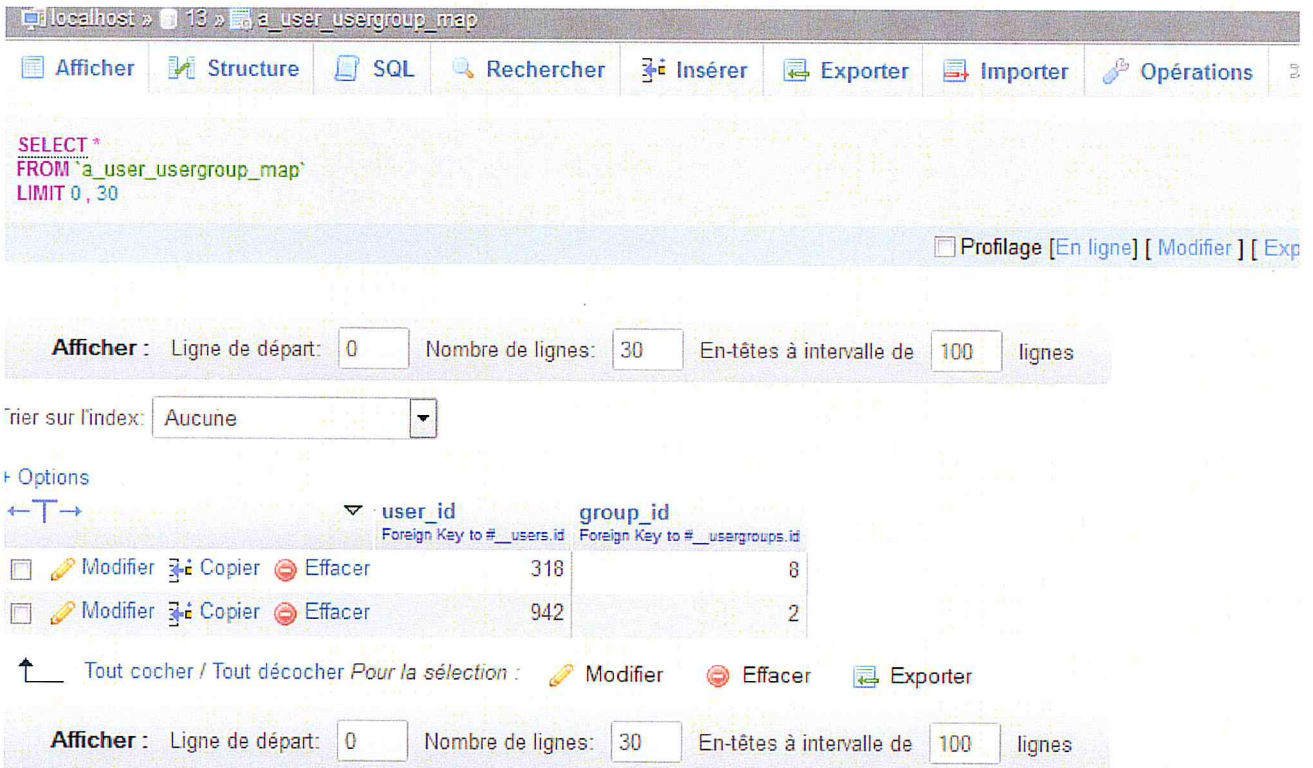


Figure 77: La table `#_user_usergroup_map` du site distant après la synchronisation sans suppression

4.1.2. Cas de suppression d'un utilisateur

On supprime un utilisateur (billel) qui va supprimer au niveau des tables `#_users` et `#_user_usergroup_map` de la base de donnée du site maitre (local).

Comme indiquer dans les figures **Figure 78** et **Figure 79**

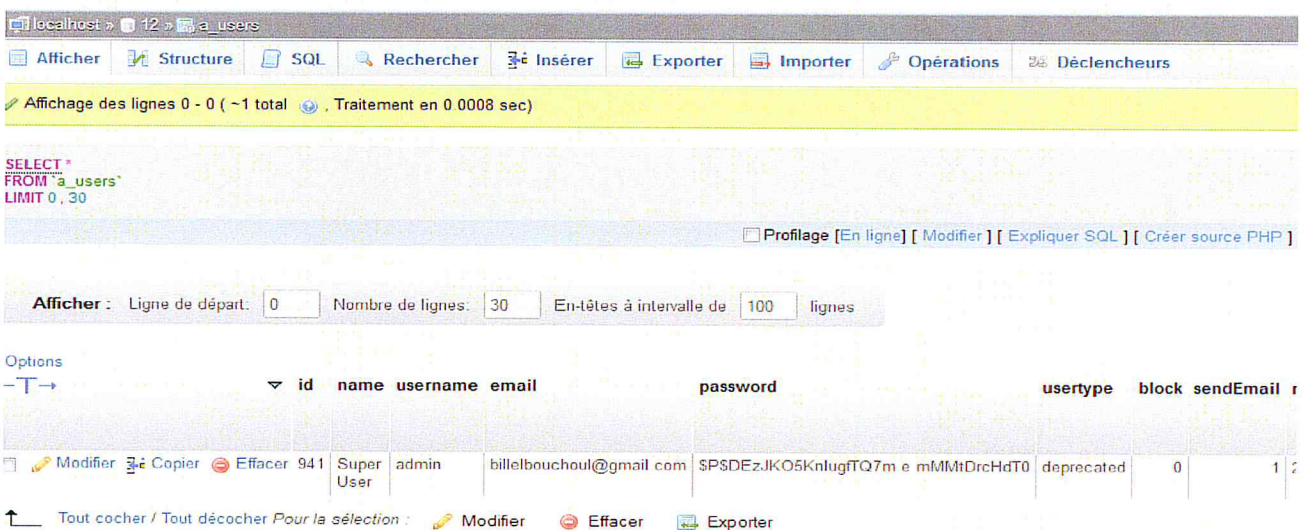


Figure 78: La table `#_users` du site local après la suppression de l'utilisateur (billel)

Chapitre 4 : Réalisation et test

The screenshot shows a database management interface with the following elements:

- Address bar: localhost » 12 » a_user_usergroup_map
- Navigation menu: Afficher, Structure, SQL, Rechercher, Insérer, Exporter, Importer
- Status bar: Affichage des lignes 0 - 0 (~1 total), Traitement en 0.0005 sec
- SQL query:

```
SELECT *  
FROM `a_user_usergroup_map`  
LIMIT 0, 30
```
- Display options: Afficher : Ligne de départ: 0, Nombre de lignes: 30, En-têtes à intervalle de 100 lignes
- Table view:

Options	
user_id	group_id
<small>Foreign Key to #__users.id Foreign Key to #__usergroups.id</small>	
941	8
- Actions: Tout cocher / Tout décocher Pour la sélection : Modifier, Effacer, Exporter

Figure 79:La table `#_user_usergroup_map` du site local après la suppression de l'utilisateur (billel)

Cette utilisateur existe dans les **tables** `#_users` et `#_user_usergroup_map` de la base de donnée du site esclave (distant)

Comme indiquer dans les figures **Figure 80** et **Figure 81**

Chapitre 4 : Réalisation et test

localhost » 13 » a_users

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Décl

Affichage des lignes 0 - 1 (~2 total) , Traitement en 0.0005 sec

```
SELECT *
FROM `a_users`
LIMIT 0, 30
```

Profilage [En ligne] [Modifier] [Expliquer S

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Trier sur l'index: Aucune

+ Options

	id	name	username	email	psd	userty
<input type="checkbox"/>	318	Super User	admin	billemouchoul@gmail.com	SP\$DX30t3JImZDOIKLZu7LTGoUM4HmZTE1	deprex
<input type="checkbox"/>	942	billem	bob	billem@gmail.com	SP\$DID568HEgbIBKOTHAdD8Z3sqU0kNTF0	

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Figure 80:La table #_users du site distant avant la synchronisation sans suppression

localhost » 13 » a_user_usergroup_map

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

```
SELECT *
FROM `a_user_usergroup_map`
LIMIT 0, 30
```

Profilage [En ligne] [Modifier] [Exp

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Trier sur l'index: Aucune

+ Options

	user_id	group_id
<input type="checkbox"/>	318	8
<input type="checkbox"/>	942	2

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Figure 81:La table #_user_usergroup_map du site distant avant la synchronisation sans suppression

Pour synchroniser les utilisateurs, On va ajouter les requêtes suivantes à la table des requêtes:

Delete from [#_users] where id =:942

Delete from [#_user_usergroup_map] where user_id =:942

Comme indiquer dans la figure **Figure 82**

Chapitre 4 : Réalisation et test

En suite en ajoute les paramètres de l'utilisateur qu'on a supprimé sur la table des paramètres, comme indiquer dans la figure **Figure 83**

The screenshot shows the sqtrack application interface. At the top, there is a navigation bar with buttons: Afficher, Structure, SQL, Rechercher, Insérer, Exporter, and Importer. Below the navigation bar, a SQL query is displayed: `SELECT * FROM 'sqtrack' LIMIT 0, 30`. To the right of the query, there is a checkbox for 'Profilage [En I]'. Below the query, there are input fields for 'Afficher': 'Ligne de départ: 0', 'Nombre de lignes: 30', and 'En-têtes à intervalle de 100 lignes'. Below these fields, there is a dropdown menu for 'Trier sur l'index:' set to 'Aucune'. Underneath, there is a section for '+ Options' with a table of operations:

	ID	Sql
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	delete from [a_users] where nomutil = :942
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	delete from [a_user_usergroup_map] where user_id = ...

At the bottom of the table, there are controls: 'Tout cocher / Tout décocher' and 'Pour la sélection:' with buttons for 'Modifier', 'Effacer', and 'Exporter'.

Figure 82:La table des requêtes pour la suppression des utilisateurs

The screenshot shows the sqparam application interface. At the top, there is a navigation bar with buttons: Afficher, Structure, SQL, Rechercher, Insérer, Exporter, and Importer. Below the navigation bar, there is a dropdown menu for 'Trier sur l'index:' set to 'Aucune'. Underneath, there is a section for '+ Options' with a table of parameters:

	PID	SQTrackID	Param
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	17	3	942
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	18	3	billem
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	19	3	bob
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	20	3	billem@gmail.com
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	21	3	\$P\$DID568HEgblBK0tHAdD8Z3sqU0kNTF0
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	22	3	
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	23	3	0
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	24	3	0
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	25	3	2014-09-19 19:52:08
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	26	3	0000-00-00 00:00:00
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	27	3	
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	28	3	{"admin_style":"","admin_language":"","language":"...
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	29	3	0000-00-00 00:00:00
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	30	3	0

Figure 83:La table des paramètres pour la suppression des utilisateurs

Chapitre 4 : Réalisation et test

Et on exécute l'algorithme de synchronisation

L'utilisateur va supprimer au niveau des tables `#_users` et `#_user_usergroup_map` de la base de donnée du site esclave, comme indiquer dans les figures **Figure 84** et **Figure 85**

The screenshot shows a database management interface for a table named `a_users`. The SQL query executed is `SELECT * FROM `a_users` LIMIT 0, 30`. The table has columns `id`, `name`, `username`, `email`, `psd`, and `usergroup_id`. The displayed data includes a row for 'Super User' with `id` 318, `username` 'admin', `email` 'billelbouchoul@gmail.com', and a long `psd` value.

id	name	username	email	psd	usergroup_id
318	Super User	admin	billelbouchoul@gmail.com	\$P\$DX30t3JlmZDOIkLZu7LTGoUM4HmZTE1	de

Figure 84:La table `#_users` après la synchronisation des utilisateurs sans suppression

The screenshot shows a database management interface for a table named `a_user_usergroup_map`. The SQL query executed is `SELECT * FROM `a_user_usergroup_map` LIMIT 0, 30`. The table has columns `user_id` and `group_id`. The displayed data includes a row with `user_id` 318 and `group_id` 8. The `user_id` column has a foreign key relationship to `#_users.id`, and the `group_id` column has a foreign key relationship to `#_usergroups.id`.

user_id	group_id
318	8

Figure 85:La table `#_user_usergroup_map` après la synchronisation des utilisateurs sans suppression

Chapitre 4 : Réalisation et test

4.2. Synchronisation des menus

4.2.1. Cas d'ajout d'un menu

On ajoute un menu (menu1) qui va enregistrer sur la table #_menu_types de la base de donnée du site maitre (local).

Comme indiquer dans la figure **Figure 86**

The screenshot shows a database management interface for a local MySQL instance. The active database is 'a_menu_types'. The SQL editor contains the query: `SELECT * FROM 'a_menu_types' LIMIT 0, 30`. Below the editor, the 'Afficher' (Display) settings are set to 'Ligne de départ: 0', 'Nombre de lignes: 30', and 'En-têtes à intervalle de 100 lignes'. The 'Index' dropdown is set to 'Aucune'. The 'Options' section shows a table with the following data:

				id	menutype	title	description			
<input type="checkbox"/>		Modifier		Copier		Effacer	1	mainmenu	Menu principal	Le menu principal du site
<input type="checkbox"/>		Modifier		Copier		Effacer	2	menu	menu 1	mon menu

At the bottom, there are controls for selection: 'Tout cocher / Tout décocher' and actions for 'Modifier', 'Effacer', and 'Exporter'.

Figure 86: La table #_menu_types après l'ajout d'un menu au site local

Ce menu n'existe pas dans la table #_menu_types de la base de donnée du site distant

Comme indiquer dans la figure **Figure 87**

Chapitre 4 : Réalisation et test

The screenshot shows a web-based SQL client interface. At the top, the browser address bar shows 'localhost' and the page title is 'a_menu_types'. The interface includes a menu with options: Afficher, Structure, SQL, Rechercher, Insérer, Exporter, Importer, and Opérations. A status bar indicates 'Affichage des lignes 0 - 0 (~1 total), Traitement en 0.0006 sec'. The SQL query entered is:

```
SELECT *  
FROM 'a_menu_types'  
LIMIT 0, 30
```

 Below the query, there is a 'Profilage' checkbox and a link '[En ligne]'. A control bar shows 'Afficher : Ligne de départ: 0, Nombre de lignes: 30, En-têtes à intervalle de 100 lignes'. Underneath, there is an 'Options' section with a table view. The table has columns: id, menutype, title, and description. One row is visible: id=1, menutype=mainmenu, title=Menu principal, description=Le menu principal du site. Below the table, there are action buttons: Modifier, Copier, Effacer, and Exporter. A final control bar at the bottom shows 'Afficher : Ligne de départ: 0, Nombre de lignes: 30, En-têtes à intervalle de 100 lignes'.

Figure 87: La table #_menu_types du site distant avant la synchronisation sans suppression

Pour synchroniser les menus, On va ajouter la requête suivante dans la table des requêtes:

```
insert into [#_menu_types](id,menutype,title,description) values (:0,:1,:2,:3)
```

Comme indiquer dans la figure **Figure 88**

En suite en ajoute les paramètres du menu qu'on a ajouté sur la table des paramètres, comme indiquer dans la figure **Figure 89**

Chapitre 4 : Réalisation et test

The screenshot shows a web-based SQL interface for a database named 'sqtrack'. The top navigation bar includes buttons for 'Afficher', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', and 'Options'. A status bar indicates 'Affichage des lignes 0 - 0 (~1 total), Traitement en 0.0004 sec'. The SQL query executed is: `SELECT * FROM 'sqtrack' LIMIT 0, 30`. Below the query, there are controls for 'Afficher' with 'Ligne de départ: 0', 'Nombre de lignes: 30', and 'En-têtes à intervalle de 100 lignes'. A table with one row is displayed, with columns 'ID' and 'Sql'. The row contains the value '5' under 'ID' and 'insert into [a_menu_types](id,menutype,title,descr...' under 'Sql'. Below the table, there are options for 'Options', a search bar, and buttons for 'Modifier', 'Copier', 'Effacer', and 'Exporter'. A second 'Afficher' control block is visible at the bottom of the screenshot.

Figure 88:La table des requêtes pour l’ajout d’un menu

The screenshot shows a web-based SQL interface for a database named 'sqparam'. The top navigation bar includes buttons for 'Afficher', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', and 'Importer'. A status bar indicates 'Affichage des lignes 0 - 3 (~4 total), Traitement en 0.0006 sec'. The SQL query executed is: `SELECT * FROM 'sqparam' LIMIT 0, 30`. Below the query, there are controls for 'Afficher' with 'Ligne de départ: 0', 'Nombre de lignes: 30', and 'En-têtes à intervalle de 100'. A 'Trier sur l'index:' dropdown menu is set to 'Aucune'. A table with four rows is displayed, with columns 'PID', 'SQTrackID', and 'Param'. The rows contain the following data: (33, 5, 2), (34, 5, menu), (35, 5, menu 1), and (36, 5, mon menu). Below the table, there are options for '+ Options', a search bar, and buttons for 'Modifier', 'Effacer', and 'Exporter'.

Figure 89:La table des paramètres pour l’ajout d’un menu

Chapitre 4 : Réalisation et test

Et on exécute le programme de synchronisation

L'utilisateur va enregistrer au niveau de la table #_menu_types de la base de donnée du site esclave (distant), comme indiquer dans la figure **Figure 90**

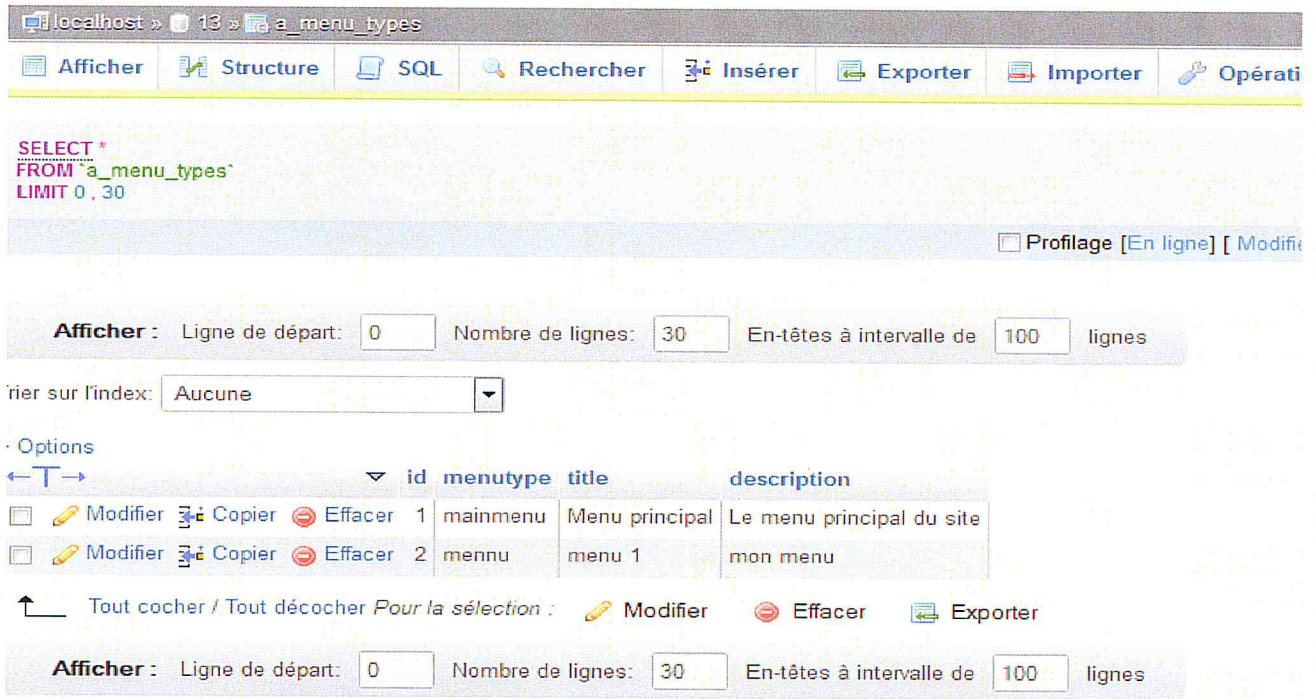


Figure 90:La table #_menu_types du site distant après la synchronisation sans suppression

4.2.2. Cas de suppression d'un menu

On supprime le menu (menu1), ce menu va supprimer au niveau de la table #_menu_types de la base de donnée du site maitre (local).

Comme indiquer dans la figure **Figure 91**

Chapitre 4 : Réalisation et test

localhost » 12 » a_menu_types

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

Affichage des lignes 0 - 0 (~1 total) , Traitement en 0.0004 sec)

```
SELECT *
FROM `a_menu_types`
LIMIT 0, 30
```

Profilage [En ligne] [

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Options

	id	menutype	title	description
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	mainmenu	Menu principal	Le menu principal du site

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Figure 91:La table #_menu_types apres la suppression du menu

Ce menu existe dans la table #_menu_types de la base de donnée du site esclave (distant)

Comme indiquer dans la figure Figure 92

localhost » 13 » a_menu_types

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

```
SELECT *
FROM `a_menu_types`
LIMIT 0, 30
```

Profilage [En ligne] [Modifi

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

rier sur l'index: Aucune

Options

	id	menutype	title	description
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	mainmenu	Menu principal	Le menu principal du site
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	menu	menu 1	mon menu

Tout cocher / Tout décocher Pour la sélection : Modifier Effacer Exporter

Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes

Figure 92:La table #_menu_types du site distant avant la synchronisation

Chapitre 4 : Réalisation et test

Pour synchroniser les menus, On va ajouter la requête suivante à la table des requêtes:

Delete from [#_menu_types] where id =:2

Comme indiquer dans la figure **Figure 93**

En suite en ajoute les paramètres de l'utilisateur qu'on a supprimé sur la table des paramètres, comme indiquer dans la figure **Figure 94**

The screenshot shows a web-based SQL client interface. The top navigation bar includes buttons for 'Afficher', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', and 'Opé'. A status bar indicates 'Affichage des lignes 0 - 0 (~1 total)' and 'Traitement en 0.0005 sec)'. The main area displays the SQL query: `SELECT * FROM `sqrack` LIMIT 0, 30`. Below the query, there are controls for 'Afficher' with fields for 'Ligne de départ: 0', 'Nombre de lignes: 30', and 'En-têtes à intervalle de 100 lignes'. An 'Options' section shows a dropdown menu set to 'ID Sql' and a list of actions: 'Modifier', 'Copier', 'Effacer', and '6'. The SQL query in the options is: `Delete from [a_menu_types] where id =:2`. At the bottom, there are more 'Afficher' controls and a 'Trier sur l'index:' dropdown set to 'Aucune'.

Figure 93:La table des requêtes pour la suppression d'uns menu

The screenshot shows a web-based SQL client interface. The top navigation bar includes buttons for 'Afficher', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', and 'Opé'. A status bar indicates 'Affichage des lignes 0 - 0 (~1 total)' and 'Traitement en 0.0005 sec)'. The main area displays the SQL query: `FROM `sqparam` LIMIT 0, 30`. Below the query, there are controls for 'Afficher' with fields for 'Ligne de départ: 0', 'Nombre de lignes: 30', and 'En-têtes à intervalle de 100 lignes'. An 'Options' section shows a dropdown menu set to 'PID SQTrackID Param' and a list of actions: 'Modifier', 'Copier', 'Effacer', and '37'. The table below shows the results of the query:

	PID	SQTrackID	Param
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	37	6	2
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	38	6	mennu
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	39	6	menu 1
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	40	6	mon menu

At the bottom, there are more 'Afficher' controls and a 'Trier sur l'index:' dropdown set to 'Aucune'.

Figure 94:La table des paramètres pour la suppression d'uns menu

Chapitre 4 : Réalisation et test

Et on exécute le programme de synchronisation

Le menu va supprimer au niveau de table # **_menu_types** de la base de donnée du site esclave, comme indiquer dans la figure **Figure 95**

The screenshot shows a web-based database management interface. At the top, the browser address bar shows 'localhost » 13 » a_menu_types'. Below the address bar is a navigation menu with buttons: 'Afficher', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', and 'Op'. A yellow status bar indicates 'Affichage des lignes 0 - 0 (~1 total), Traitement en 0.0007 sec'. The main area displays a SQL query: 'SELECT * FROM `a_menu_types` LIMIT 0, 30'. Below the query is a 'Profilage [En ligne]' checkbox. A control bar shows 'Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes'. Below this is an 'Options' section with a table view. The table has columns 'id', 'menutype', 'title', and 'description'. There is one row with values: '1', 'mainmenu', 'Menu principal', and 'Le menu principal du site'. Below the table are action buttons: 'Modifier', 'Copier', 'Effacer', and 'Exporter'. A 'Tout cocher / Tout décocher' option is also present. At the bottom, another control bar shows 'Afficher : Ligne de départ: 0 Nombre de lignes: 30 En-têtes à intervalle de 100 lignes'.

Figure 95: La table # **_menu_types** après la synchronisation des menus sans suppression

Pour la gestion des articles et gestion des catégories en applique les mêmes étapes on change seulement les paramètres des requêtes SQL.

Conclusion Général

Conclusion général

Conclusion Générale

Le thème de ce mémoire est la conception et la réalisation d'une application de synchronisation d'un site Joomla local-Joomla distant qui permet la synchronisation des données des articles, des menus et des catégories et des utilisateurs qui répond dans l'intérêt de garder une image miroir du site distant localement.

Au chapitre trois, nous avons évoqué différents problèmes liés à la synchronisation. La plupart sont difficile à prendre en considération. Ils pourraient induire d'ajouter des tables aux bases de Joomla pour historier les opérations qui ont eu lieu.

Le travail de ce mémoire a consisté plus en un défrichage d'un domaine plutôt vierge qu'en une application finie. N'en témoigne que l'absence d'une pareille application sur le Web, ce qui en dit long sur la problématique. Il a permis d'évoquer un certain nombre de problèmes.

Le travail pourrait être raffiné en essayant d'apporter des solutions aux problèmes cités mais aussi en prenant en compte d'autres objets liés au contenu tel les commentaires.

Plus loin pourrait aboutir le travail s'il devait prendre en considération les aspects apparence, fonctionnalités non natives. Il devrait alors utiliser des fonctions de transfert ftp au moins pour une synchronisation de fichiers locaux et distants, en plus des changements entre les deux bases.

Références

[DOC_01] : « Joomla! 2.5 Le Guide Pour Débutant », Hagen Graf, Milena Mitova, édition 1.

[DOC_02] : Conception et réalisation d'un générateur de charte graphique pour Joomla
<<TouatiAbdelhak Et Zitoun Abderrazak 2012 2013 >>.

[DOC_03] : Thierry Duboise <<Tous pour réussir un site web, tome 2, Septembre 2010>>.

[DOC_04] : «Advanced Joomla ! », Dan Rahmel, Apress, édition 1.

[DOC_05] : «Introduction pratique aux bases de données relationnelles», Andreas Meier, Springer, édition 2.

[DOC_06] : «high performance MySQL», Jeremy D. Zawodny & Derek J. Balling, O'Reilly, édition 1.

[DOC_07] : «Pro Sync Framework», Rituraj Singh and Joydip Kanjilal, Apress, édition 1.

[WEB_01] : http://docs.joomla.org/Portal:Using_the_Database

[WEB_02] : <http://docs.joomla.org/Article/fr>

[WEB_03] : <http://www.bluesharksolution.com/services/web-hosting/joomla-web-design-development-company.html>

[WEB_04] : <http://www.techterms.com/definition/visualbasic>

[WEB_05] : <http://www.wampserver.com/>

[WEB_O6] : http://greg.rubyfr.net/pub/?page_id=20

[WEB_O7] : <http://www.developpez.net/forums/d299885/php/scripts/cms/joomla/mcd-joomla/>