

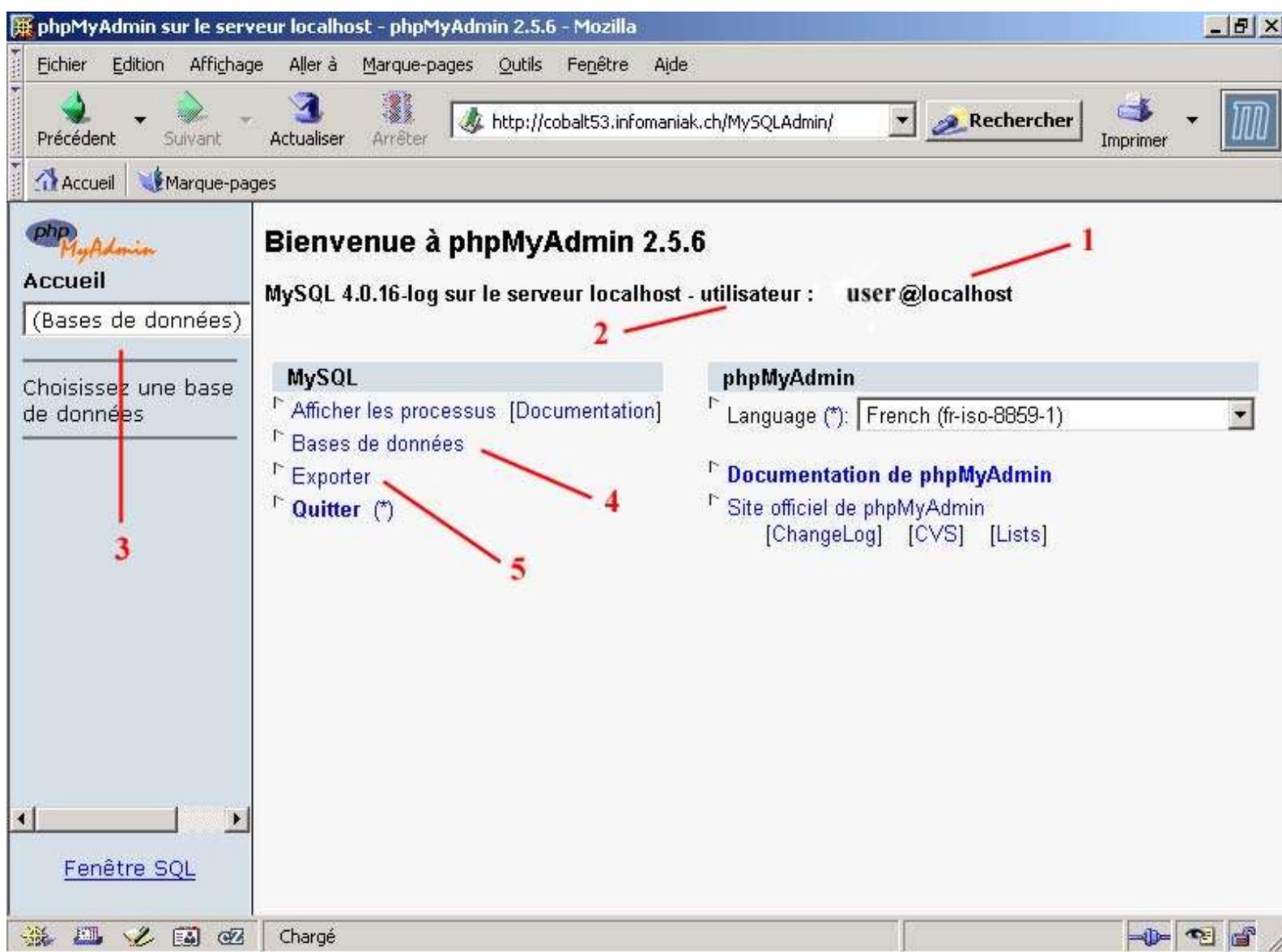
PhpMyAdmin

Je vais vous présenter, dans ce tutorial, les fonctionnalités de base de PhpMyAdmin: Création de bases de données, création de tables, modification de tables, insertion de données, exportation et importation de données.

PhpMyAdmin est donc une interface pour gérer une (des) base(s) de données, cela vous évitera de le faire par le biais de PHP ou de commandes MySQL qui sont difficiles à intégrer.

Ecran d'accueil

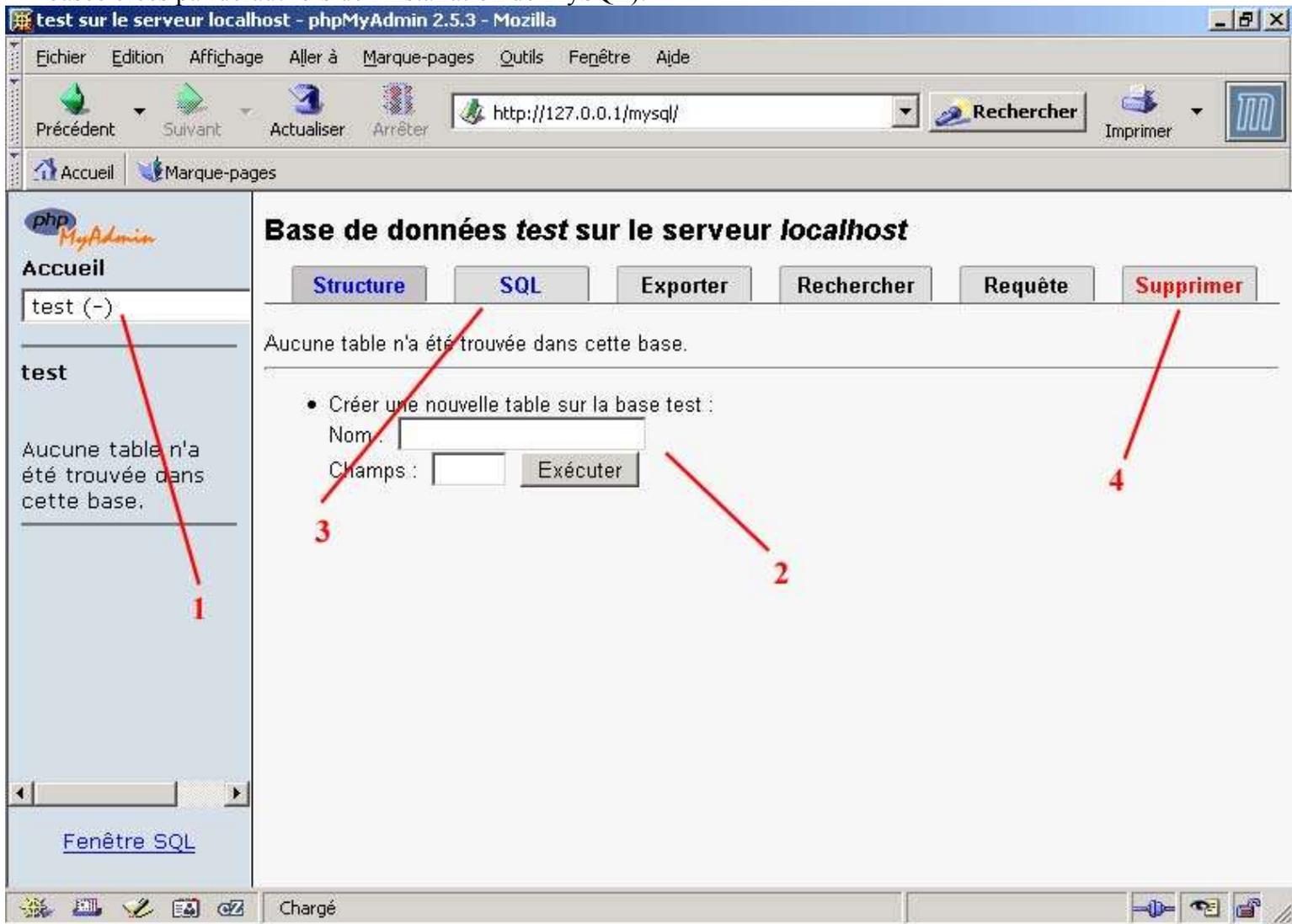
Lorsque vous vous connectez à PhpMyAdmin, si vous avez plusieurs bases de données, vous vous trouverez face à cet écran (si vous n'avez qu'une base de données, reportez vous à la prochaine image).



Voici donc à quoi correspondent les différents numéros:

- 1: Nom de l'utilisateur ([utilisateur@serveur](#))
- 2: Nom du serveur
- 3: Cette liste vous permet d'accéder à une base de données
- 4: Afficher la liste des bases de données
- 5: Permet d'exporter une (des) base(s) de données, cela permet de les réimporter en cas de besoin. Il s'agit donc d'une sauvegarde.

Choisissez une base de données à gauche, voici à quoi peut ressembler votre écran (cette base est une des deux basées créées par défaut lors de l'installation de MySQL):



- 1: Nom de la base (nombre de table(s))
- 2: Formulaire pour la création d'une nouvelle table
- 3: Permet d'exécuter des requêtes MySQL
- 4: Permet de supprimer la base courante

Création d'une table

Nous allons étudier la création d'une nouvelle table: La table pour la création d'un livre d'or. Nous allons créer les champs suivants:

- id (identifiant unique qui représentera chaque message) - entier
- auteur (auteur du message) – texte: 32 caractères
- date (date du message) – datetime (aaaa-mm-jj hh:mm:ss)
- message (corps du message) – texte

Remplissez le formulaire « Créer une nouvelle table » comme ceci:

- Nom: livre_or (nom de la table, maximum 64 caractères)
- Champs: 4 (nombre de champs)

Après avoir cliqué sur « Exécuter », vous allez vous retrouver face à un tableau de 4 lignes, chacune représentant un champ de votre future table.

- Champ: Nom du champ (maximum 64 caractères)
- Type: Type de valeur, à choisir parmi les suivants:

CHAR: Chaîne de caractères de longueur fixe. Longueur maximale comprise entre 1 et 255
VARCHAR: Chaîne de caractères de longueur variable. Longueur maximale comprise entre 1 et 255
TINYTEXT: Texte de longueur variable pouvant aller jusqu'à 255 caractères
TEXT: Texte de longueur variable pouvant aller jusqu'à 65535 caractères
MEDIUMTEXT: Texte de longueur variable pouvant aller jusqu'à 16777215 caractères
LONGTEXT: Texte de longueur variable pouvant aller jusqu'à 4294967295 caractères
BLOB: Les types **TINYBLOB**, **BLOB**, **MEDIUMBLOB** et **LOBLOB** sont identiques à leurs homologues **TEXT** à la différence que les recherches sur un type **BLOB** sont sensibles à la casse

TINYINT: Entier codé sur un octet . Valeurs: de -128 à 127 (0 à 255 en unsigned) ([char](#))
SMALLINT: Entier codé sur 2 octets. Valeurs de -32768 à 32767 (0 à 65535 en unsigned) ([small int](#))
MEDIUMINT: Entier codé sur 3 octets. Valeurs de -8388608 à 8388607 (0 à 16777215 en unsigned)
INT: Entier codé sur 4 octets. Valeurs de -2147483648 à 2147483647 (0 à 4294967295 en unsigned) ([int](#))
BIGINT: Entier codé sur 8 octets. Valeurs de -9223372036854775808 à 9223372036854775807 (0 à 18446744073709551615 en unsigned) ([__int64](#))

FLOAT: Décimal en précision simple, codé sur 4 octets. Valeurs $-3.402823466*10^{38}$ à $-1.175494351*10^{-38}$ et de $1.175494351*10^{-38}$ à $3.402823466*10^{38}$ ([float](#))

DOUBLE: Décimal en double précision, codé sur 8 octets. Valeurs $-1.7976931348623157*10^{308}$ à $-2.2250738585072014*10^{-308}$ et de $2.2250738585072014*10^{-308}$ à $1.7976931348623157*10^{308}$ ([double](#))

DECIMAL: Décimal sous forme de chaîne de caractères (la taille est variable: 1 chiffres => 1 caractère)

DATE: Date au format aaaa-mm-jj

DATETIME: Date au format aaaa-mm-jj hh:mm:ss

TIMESTAMP: Sans informations supplémentaires, équivalent au **DATETIME** mais sans les séparateurs. Possibilité de créer un **TIMESTAMP** incomplet (par exemple AAMMJJHH). **Remarque**: En PHP, un **TIMESTAMP** est un entier représentant le nombre de secondes écoulées depuis le 1^{er} janvier 1970, tandis qu'en MySQL, comme nous venons de le voir, il s'agit d'une date formatée.

TIME: Heure au format hh:mm:ss

YEAR: Année au format aaaa

ENUM: Permet de représenter un ensemble défini de valeurs choisies lors de la création de la table (peut aussi prendre la valeur nulle). On peut l'associer à un ensemble de champs radios: Une valeur parmi un ensemble est choisie.

SET: Identique que le type **ENUM** à la différence que l'on peut choisir plusieurs des valeurs possibles. On peut l'associer à un ensemble de checkboxes.

(en bleu, quand c'est possible, correspondance avec le C++)

- Taille/Valeur:
 - Dans le cas d'entiers, cela permet de spécifier le nombre de chiffres: INT(3) ne permet que de -999 à 999
 - Dans le cas de **VARCHAR** ou de **CHAR**, cela permet de spécifier le nombre de caractères maximal
 - Dans le cas de **TIMESTAMP**, cela spécifie sa longueur: **TIMESTAMP**(10) --> AAAAMMJJHH
 - Dans le cas de **SET** ou **ENUM**, cela permet de spécifier les valeurs possibles. Pour un champ qui peut prendre les valeurs 1, 5, n : '1', '5', 'n'
- Attributs:
 - BINARY**: Pour les chaînes de caractères: Rend les recherches sensibles à la casse
 - UNSIGNED**: Nombre non-signé
 - UNSIGNED ZEROFILL**: Nombre non-signé avec remplissage des 0 non significatifs (9 --> 00000009, par exemple)

Explication sur le terme 'unsigned':

1 octet correspond à 8 bits, c'est-à-dire qu'un **TINYINT**, codé sur un octet, est représenté comme ceci pour la machine: 00000000 . Le nombre de possibilités total pour un nombre donné de bits est égal à $2^{\text{nombre de bits}}$. Ainsi, avec 8 bits, on peut avoir $2^8 = 256$ possibilités.

Utiliser un nombre 'unsigned', c'est utiliser un nombre sans se préoccuper de son signe, on peut donc

représenter les nombres de 0 à 255.

Si l'on utilise un nombre signé, un bit est réservé pour le signe, on peut donc représenter les valeurs de -128 à 127.

- Null:
null: Le champ peut avoir la valeur nulle
not nul: Le champ ne peut pas avoir la valeur nulle
- Défaut:
Représente la valeur par défaut: Si on est spécifié pas de valeur lors de l'insertion d'une ligne, cette valeur sera utilisée.
- Extra:
auto_increment: Lors de l'insertion, on ne spécifie pas de valeur pour ce champ qui prendra automatiquement une valeur: 1, 2, 3, 4, 5 ... Utile pour les index dont on ne veut pas se préoccuper.
- Primaire:
Crée une clé primaire: Il s'agit d'un index. Il ne pourra pas y avoir de doublons avec cette option. C'est utile pour identifier chaque enregistrement de manière unique.
- Index:
Crée un index, tout comme **Primary** sauf qu'il peut y avoir des doublons.
- Unique:
Exclu les doublons pour la colonne créée avec l'option **Unique**.

Explication sur le terme 'Index':

Les informations concernant une table MySQL sont stockées dans 3 fichiers différents:

- table.frm : Contient les informations sur la structure de la table
- table.myi : Contient les informations relatives aux index
- table.myd: Contient les données de la table

Le fichier myi contient les index ainsi que les enregistrements auxquels ils se réfèrent. Cela permet d'accéder à un enregistrement sans devoir lire toute la table.

Certaines requêtes bien optimisées peuvent mettre en jeu que les index, ce qui permet de ne pas ouvrir le fichier qui contient les données et par conséquent de gagner du temps.

Voici ce que peut donner notre table livre_or:

Base de données test - Table livre_or sur le serveur localhost

Champ	Type [Documentation]	Taille/Valeurs*	Attributs	Null	Défaut**	Extra	Primaire	Index	Unique	...	Texte entier
id	SMALLINT		UNSIGNED	not null		auto_increment	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>
auteur	VARCHAR	32		not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>
date	DATETIME			not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>
message	TEXT			not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>

Commentaires sur la table :

Table en format :

Sauvegarder

Une fois votre table créée, la requête SQL qui sert à créer cette table est affichée (d'une manière générale, PhpMyAdmin affiche la requête SQL d'une action après l'exécution de celle-ci):

```
CREATE TABLE `livre_or` (  
  `id` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `auteur` VARCHAR( 32 ) NOT NULL ,  
  `date` DATETIME NOT NULL ,  
  `message` TEXT NOT NULL ,  
  PRIMARY KEY ( `id` )  
);
```

En dessous, vous avez la possibilité de modifier la structure de votre table (ajouter ou supprimer des champs, modifier des champs, ...).

A gauche, dans le menu, vous pouvez constater que la nouvelle table *livre_or* apparaît dans votre base de données.

En haut, vous avez plusieurs liens (s'il n'apparaissent pas, cliquez sur le nom de votre nouvelle table, à gauche):



- Structure:
Page courante: Permet d'afficher et de modifier la structure de la table
- Afficher:
Quand la table contiendra des enregistrements, vous pourrez utiliser ce lien pour les afficher
- SQL:
Permet d'exécuter une requête SQL
- Rechercher:
Recherche d'un enregistrement parmi ceux existants
- Insérer:
Insertion d'un enregistrement
- Exporter:
Permet d'exporter la base pour une ré-importation future (sauvegarde par exemple)
- Opérations:
Permet diverses actions sur la table dont la modification de la valeur de l'auto_increment
- Vider:
Efface tous les enregistrements contenus dans la table
- Supprimer:
Supprime la table

Insertion d'un enregistrement

Cliquez sur le lien *Insérer*

Vous allez obtenir un tableau contenant 5 colonnes:

- Champ:
Nom du champ
- Type:
Type de valeur du champ
- Fonction:
Valeurs prédéfinies et fonctions:
 - ASCII(str): Renvoie le code ASCII du premier caractère de str
 - CHAR(n): Retourne le caractère représenté par le code ASCII n
 - CURDATE: Retourne la date courante au format AAAA-MM-JJ ou AAAAMMJJHH, selon le type du champ
 - CURTIME: Retourne l'heure courante au format HH:MM:SS ou HHMMSS, selon le type du champ
 - ENCRYPT(str): Crypte la chaîne str avec crypt()
 - FROM_DAYS(n): Retourne la date correspondant à n jours après le jour 0
 - FROM_UNIXTIME(n) Retourne le timestamp passé en paramètre au format AAAA-MM-JJ HH:MM:SS ou AAAAMMJJHHMMSS selon le type de champ
 - LAST_INSERT_ID: Dernière valeur générée automatiquement pour un champ auto_increment
 - LCASE(str): Passe la chaîne str en minuscule
 - MD5(str): Crypte la chaîne str avec MD5 (entier hexadécimal de 32 caractères)
 - NOW: Retourne la date courante au format: AAAA-MM-JJ HH:MM:SS ou AAAAMMJJHHMMSS, selon le type du champ (peut même ne retourner qu'une partie, notamment pour un entier trop petit).
 - PASSWORD(str): Retourne le password à partir de la chaîne str (cette fonction est utilisée pour créer les passwords MySQL)

- RAND: Retourne un nombre aléatoire à virgule flottante compris entre 0 et 1
- SOUNDEX(str): Valeur Soundex de le chaîne str
- TO_DAYS(date): Retourne le nombre de jours depuis la date 0
- UNIX_TIMESTAMP(date): Retourne le timestamp unix de la date passée en paramètre
- USER: Retourne le nom d'utilisateur et l'hôte courant (par exemple: root@localhost)
- WEEKDAY(date): Retourne le jour se la semaine correspondant à date (Lundi = 0, Mardi = 1, ...
Dimanche = 6)
- Null:
Si c'est options est activée (si le champ le permet) la valeur sera Null
- Valeur:
Valeur du champ (dans le cas d'utilisation de fonctions MySQL (vues au dessus, paramètre de la fonction))

Voici ce que peut donner l'insertion de données pour notre table livre_or:

Champ	Type	Fonction	Null	Valeur
id	smallint(5) unsigned	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
auteur	varchar(32)	<input type="text"/>	<input type="checkbox"/>	R@f
date	datetime	<input type="text"/>	<input type="checkbox"/>	2004-15-08 17:49:41
message	text	<input type="text"/>	<input type="checkbox"/>	Hello! Voici le premier message à enregistrer dans le livre d'or... @++ R@f