



Initiation au SGBD MySQL

Introduction

(1of18)

SQL

("Structured Query Language" en anglais, ce qui signifie Langage d'interrogation structuré, ou en simplifiant **langage de requêtes**) est un langage de définition et de manipulation de bases de données **relationnelles**.

Il définit un **standard** qui a été **normalisé**(ANSI); chaque produit SQL **réalise différemment** et plus ou moins strictement la norme.

MySQL

Est un **serveur** de base de données exploitant un langage qui dérive directement de SQL, en **reprenant la syntaxe**, mais **pas toutes les fonctionnalités** (clés imbriquées, clés étrangères...)



Introduction

(2of18)

Historique de SQL

1970

Découverte de l'algèbre relationnelle (E.F.Codd)

jusqu'en 1977

SEQUEL (IBM)

->SEQUEL/2(serveur Système d'IBM)

1981

Relational Software

Oracle (premier SGBD relationnel implémentant SQL)

1986

Normalisation de SQL

->SQL/86

1992

Nombreuses améliorations->modification de la norme

->SQL/92=SQL2

1999

Expressions rationnelles, requêtes récursives, déclencheurs, types non-scalaires et quelques fonctions orientées objet

->SQL-99ouSQL3

2003

Fonctions pour la manipulation XML, «window functions»,ordres standardisés et colonnes avec valeurs auto-produites

->SQL:2003

Introduction

(3of18)

Les sous-langages de SQL

LDD : Langage de Définition des Données

(DDL: *Data Definition Language* en anglais)

-> permet de **créer, modifier ou supprimer** les définitions des tables de la base

De **modifier la structure** de la base de données

LMD: Langage de Manipulation des Données

(DML: *Data Manipulation Language*)

-> permet de **manipuler** les données dans les tables

De **consulter/modifier le contenu** de la base de données

LCD : Langage de Contrôle des Données

(DCL: *Data Control Language*)

-> permet de **gérer les protections d'accès** aux tables

De **gérer les privilèges** (les droits des utilisateurs)

LCT: Langage de Contrôle des Transactions

(TCL: *Transaction Control Language*)

-> permet de **gérer les interruptions en cours de traitement**

Et permettre de remettre la base de données dans un état cohérent

De **gérer les transactions**, c'est-à-dire rendre automatique divers ordres enchaînés en séquence

SQLprocédural,

Dont *PSM Persistent Stored Module, CLICall Level Interface*

Et SQL embarqué (*Embedded SQL* en anglais),...

-> permet d'**interfacer SQL** avec des langages hôtes

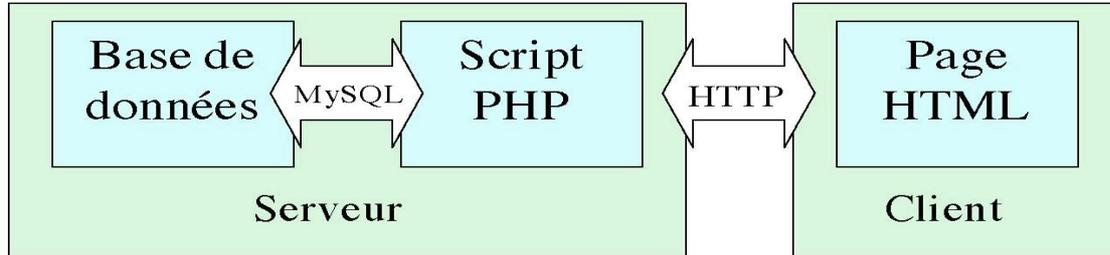
Exploiter un ensemble d'outils

MySQL pour le Web:Architecture3-tiers

(4of18)

MySQL est très utilisé pour le développement des sites Web dynamiques, et proposé par des hébergeurs gratuits comme **free**, **lycos**, **OVH**, ...

Mais il n'est pas le seul (**postgreSQL**, **msql**, **SQLplus**,...) et peut être utilisé en dehors du contexte des sites Web.



Administration d'un serveur

(5of18)

La communication avec un serveur nécessite
Une **phase de connexion**, avec identifiant et mot de passe
Il existe en gros **3 niveaux de permission** pour les utilisateurs:

Les Administrateurs

Peuvent agir sur
Les **bases de données**, et donc
l'accès et l'existence
des tables

Les Concepteurs

peuvent agir sur
les **structures des tables**,
et donc les colonnes

les Utilisateurs

peuvent agir sur
les **contenus des tables**,
et donc les lignes

Chaque niveau exploite principalement un des sous-langages SQL

LCD, LCT,
SQL procédural

LDD

LMD

Administration d'un serveur

(6of18)

Tâches des Administrateurs

- limiter le nombre de requêtes/mises-à-jour/connexions par jour et par utilisateur
- démarrer/arrêter le serveur
- créer/gérer les utilisateurs
- accéder à toutes les tables, et bloquer l'utilisation de certaines tables
- sauvegarder/restaurer tout ou partie des bases de données
- créer des processus pour automatiser des tâches

Donc agir sur les **tables**

Administration d'un serveur

(7of18)

Tâches des Concepteurs/Réalisateurs

- créer/supprimer des tables, et des tables temporaires
- modifier leurs structures, dont changer de statut à des attributs (index)
- exécuter des processus

Donc agir sur les **colonnes**

Tâches des Utilisateurs/Visiteurs

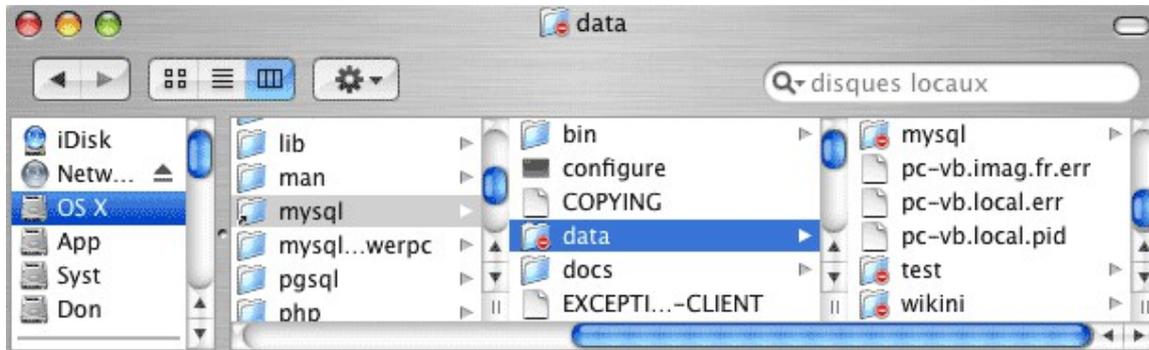
- voir le contenu des tables auxquelles ils sont autorisés, même sur plusieurs tables à la fois (jointures)
- ajouter/supprimer des valeurs à certaines tables
- ...

Donc agir sur les **lignes**

Administration d'un serveur

(8of18)

Les utilisateurs et leurs niveaux de permission sont mémorisés par le serveur dans une base de données parmi celles qu'il gère



Algèbre relationnelle

(9of18)

Rappels de vocabulaire

■ Domaine :

Ensemble de valeurs d'un même type, pouvant toutes caractériser/qualifier une propriété récurrente.

■ Table relationnelle:

Table à 2 dimensions, chaque colonne étant caractérisée par un nom.

■ Relation(Table relationnelle):

Sous ensemble du produit cartésien d'une liste de domaines.

Le résultat est un tableau à deux dimensions dont les colonnes correspondent aux domaines, et sont nommées, et dont les lignes contiennent des tuples.

■ Attribut:

Une colonne d'une relation, caractérisé par un(son)nom.

■ tUple :

Liste des valeurs d'une ligne d'une table relationnelle.

Algèbre relationnelle

(10of18)

Les opérations de base

- **Projection :** Sélection d'un ou plusieurs attributs d'une relation.
Exemple: n'afficher que les colonnes `Nom` et `Prenom` d'une table `Personnes`.
- **Jointure:** Création d'une nouvelle relation à partir de 2 ou plusieurs autres en prenant comme pivot 1 ou plusieurs attributs.
Exemple: concaténer deux tables d'adresses en fonction du nom de famille pour obtenir dans une seule table les informations provenant des 2 tables (c'est typiquement du recoupement de fichiers).
- **Sélection:** Sélection de tous les tuples ou bien seulement d'une partie en fonction de critères de sélection qui portent sur les valeurs des attributs.
Exemple: n'afficher que les lignes de la table `Personnes` qui vérifient la condition suivante:
le nom ne commence pas par la lettre 'C'.

Cette algèbre est facilement possible avec les commandes de MySQL

```
(SELECT...FROM...WHERE...)
```

Données SQL

(11of18)

Types des données

Chaque attribut peut porter des valeurs d'un seul et même type.

Les types utilisables peuvent nécessiter plus ou moins de place en mémoire.

Pour pouvoir ajuster au plus près le type des données aux besoins des situations réelles, MySQL supporte des types très différents :

- Nombre entier signé ou non (température, quantité commandée, âge)
- Nombre à virgule (prix, taille)
- Chaîne de caractères (nom, adresse, article de presse)
- Date et heure (date de naissance, heure de parution)
- Énumération (une couleur parmi une liste prédéfinie)
- Ensemble (une ou des monnaies parmi une liste prédéfinie)

DonnéesSQL

(12of18)

Exemple du type entier

nom	Borne inférieure	Borne supérieure
TINYINT	-128	127
TINYINTUNSIGNED	0	255
SMALLINT	-32768	32767
SMALLINTUNSIGNED	0	65535
MEDIUMINT	-8388608	8388607
MEDIUMINTUNSIGNED	0	16777215
INT*	-2147483648	2147483647
INT*UNSIGNED	0	4294967295
BIGINT	-9223372036854775808	9223372036854775807
BIGINTUNSIGNED	0	18446744073709551615

INTEGER est un synonyme de INT.

UNSIGNED permet d'avoir un type non signé.

ZEROFILL: remplissage des zéros non significatifs.

Données SQL

(13of18)

Exemple du type caractère

nom	longueur
CHAR(M)	Chaîne de taille fixée à M, où $1 < M < 255$, complétée avec des espaces si nécessaire.
CHAR(M)BINARY	Idem, mais insensible à la casse lors des tris et recherches.
VARCHAR(M)	Chaîne de taille variable, de taille maximum M, où $1 < M < 255$, complété avec des espaces si nécessaire.
VARCHAR(M)BINARY	Idem, mais insensible à la casse lors des tris et recherches.
TINYTEXT	Longueur maximale de 255 caractères.
TEXT	Longueur maximale de 65535 caractères.
MEDIUMTEXT	Longueur maximale de 16777215 caractères.
LONGTEXT	Longueur maximale de 4294967295 caractères.
DECIMAL(M,D)	Simule un nombre flottant de D chiffres après la virgule et de M chiffres au maximum. Chaque chiffre ainsi que la virgule et le signe moins (pas le plus) occupe un caractère.
NUMERIC(M,D)	

Requêtes SQL

(14of18)

Niveau Utilisateur

Agissant sur le **contenu** d'une table

SELECT

```
SELECT `date`, `val` FROM  
`essai`  
LIMIT 0, 30
```

Et les autres...(FILE)

SELECT, INSERT, UPDATE, DELETE, FILE

INSERT

(INSERT...)

UPDATE

(UPDATE...)

DELETE

(DELETE...)

Requêtes SQL

(15of18)

Niveau Concepteur

Agissant sur la **structure** d'une table

CREATE

```
CREATE TABLE `essai` (  
  `no` SMALLINT NOT NULL  
  
  AUTO_INCREMENT PRIMARY KEY,  
  `val` VARCHAR(32) NULL,  
  `date` DATE NULL)  
ENGINE=MYISAM;
```

ALTER

```
ALTER TABLE `essai` ADD `test` VARCHAR  
( 64 ) NOT NULL AFTER `val` ;
```

DROP

(DROP...)

CREATE, ALTER, INDEX, DROP, CREATE TEMPORARY TABLES, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE

Requêtes SQL

(16of18)

Niveau Administrateur

Agissant sur les **bases de données** et les **utilisateurs**

CREATEUSER

```
(CREATE USER...)
```

SHOWDATABASES

```
(SHOW DATABASES...)
```

LOCKTABLES

```
(LOCK TABLES...)
```

SHOW DATABASES, LOCK TABLES, REFERENCES, REPLICATION CLIENT, REPLICATION SLAVE, CREATE USER

Requêtes SQL

(17of18)

Niveau Administrateur

Agissant sur le **serveur** lui-même

SHUTDOWN

(SHUTDOWN...)

RELOAD

(RELOAD...)

GRANT

(GRANT...)

GRANT, SUPER, PROCESS, RELOAD, SHUTDOWN

Php MyAdmin

(18of18)

Pour les tâches de construction des bases de données en MySQL, généralement on utilise un ensemble de pages Web résultant de l'exécution de scripts en PHP.

Php MyAdmin est issu d'un développement "*open source*" (logiciels libres). Il permet de spécifier des requêtes MySQL en remplissant des formulaires HTML

Démo

Voir directement sur "site"

- chez l'hébergeur "**free**" : <http://sql.free.fr>