Support de cours établi par
Denis Szalkowski
Formateur Consultant

Sommaire

Présentation

Nature du produit

MySql est un système de gestion de base de données relationnel et transactionnel en utilisant InnoDb dont le propriétaire est Oracle. MySql devrait remplacer InnoDb par Falcon.
Les performances du nouveau moteur sont très en deçà de InnoDb.
http://www.mysqlperformanceblog.com/2007/01/08/innodb-vs-myisam-vs-falcon-benchmarks-part-1/
Il s'exécute sur un serveur sous forme d'un service sous Windows ou d'un démon sous Linux. C'est un produit qui n'a rien à voir avec un produit tel qu'Access qui n'autorise que le partage de fichiers Mdb.

Compatibilité

Vous pouvez l'installer indifféremment sous Linux ou sous Windows, toutes versions confondues. Le port Tcp qu'il utilise par défaut est le port 3306.

Où se le procurer ?

Vous pouvez le télécharger à partir de l'adresse http://www.mysql.com . Il est sous licence GPL (General Public License), ce qui signifie que vous pouvez le distribuer comme bon vous semble. En mode gratuit, ses conditions d'utilisation demeurent toutefois restreintes : le code qui utilise MySql doit être en Gpl.
L'éditeur met à disposition la version Enterprise au format binaire unqiuement pour ses clients payants.

Limitations

Ce produit présente quelques limitations telles que la non prise en charge des contraintes de vérification lors de la création des tables.
En terme de volumétire, les tables sont limitées à 8TB.

Fonctionnalités

Plans for some of the most requested features are summarized in the following table.

Union	4.0
Sous requête	4.1
r-trees	4.1 (MyISAM)
Procédures stockées	5.0
Vues	5.0
Curseurs	5.0
Clés étrangères	5.1 (depuis la 3.23 avec InnoDB)
Triggers	5.1
Full outer join	5.1
Contraintes	5.1

Installation MySql et configuration

Prototypage du fichier de configuration

Small	<64Mo
Medium	<128Mo
Large	<512Mo
Huge	<2 Go
Heavy	<4Go

Sous Windows

Par défaut, MySql s'installe dans c:\Program files\MySql.

my.cnf ou my.ini

Les informations de configuration du logiciel sont stockées dans c:\windows\my.ini ou dans c:\my.cnf et s'exécutent dans cette priorité :

1. %SystemRoot%/my.ini
2. C:\my.cnf

Dans le cas où vous installez MySql ailleurs que dans le répertoire C:\MySql, n'oubliez pas de créer un fichier C:\MY.CNF, copie du fichier MY.INI en n'oubliant pas d'indiquer le répertoire où se trouve le code ainsi que les bases de données.
Dans l'exemple ci-dessous, MySql a été installée sur le lecteur D.

```
[mysqld]
basedir=d:/MySql/
datadir=f:/appli/data
```

Service Windows Nt/ Xp / 2000

Le service MySql se lance à partir de c:/mysql/bin/mysqld-nt.exe .

Lancement manuel Windows 9x/Nt /2000/Xp

Vous pouvez indifféremment lancer le démon à partir de c:/mysql/bin/winmysqladmin.exe ou bien c:/mysql/bin/mysqld-opt.exe

Odbc

Lors de l'installation du pilote, n'oubliez pas de cliquer sur MySql dans la liste des pilotes.
Sous Nt/2000, pensez à faire une sauvegarde des trois fichiers qui suivent :

```
ctl3d32.dll
ds32gt.dll
```

```
odbc32gt.dll
```

Organisation physique des bases de données et des tables

Sous MySql, une base de données est un sous-répertoire du répertoire Défini par la variable datadir dans le fichier my.ini ou my.cnf.
Les tables sont organisées de la façon suivante :

- un fichier frm qui contient la définition des tables
- un fichier myd qui contient les données
- un fichier myi qui contient les index

Sous Linux

Système de fichiers

ReiserFS, XFS et JFS sont supérieurs au système ext2-ext3.
Vous disposez de deux défragmenteurs sous Linux : shake et defrag.

my.cnf

Le fichier de configuration est, dans cet ordre, exécuté à partir de :

1. /etc/my.cnf
2. datadir/my.cnf
3. ~/.my.cnf

Localisations des fichiers de configuration sous CentOs

/etc/my.cnf
/usr/share/mysql/
/usr/share/doc/mysql-server-x.x.xx/
/var/lib/mysql

Configuration à chaud

set GLOBAL
set SESSION

Quelques lignes de configuration

```
datadir = /data/myisam

log-bin = /usr/local/mysql/data/repl/bin-log

innodb_data_home_dir=

innodb_data_file_path=/dev/sdb1:18Graw;/dev/sdc1:18Graw
innodb_log_group_home_dir = /usr/local/mysql/data/iblog
```

innodb_log_arch_dir = /usr/local/mysql/data/iblog

Logger l'activité MySql

Editez le fichier /etc/my.cnf -> log = /var/log/mysql/mysql.log

Redémarrez le service MySql -> /etc/init.d/mysql restart

Affichez le contenu du fichier de log -> tail -f /var/log/mysql/mysql.log

Informations

```
SHOW VARIABLES;
```
SHOW PROCESSLIST \G
SHOW FULL PROCESSLIST.
SHOW STATUS
SHOW INNODB STATUS
SHOW ENGINE STATUS
SHOW GRANTS FOR user
```
SHOW INDEXES FROM access_jeremy_zawodny_com \G
```

Consomation mémoire

min_memory_needed = global_buffers + (thread_buffers * max_connections)

thread_buffers sort_buffer

myisam_sort_buffer

read_buffer

	join_buffer
	read_rnd_buffer
global_buffers	key_buffer
	innodb_buffer_pool
	innodb_log_buffer
	innodb_additional_mem_pool
	net_buffer

Les outils

Toad for MySql et SQLYog

Les deux produits sont gratuits.

Test des injections MySql

Vous disposez à ce jour et à ma connaissance de deux produits :

- MySql Power Injector en .Net
- WebGoat en J2EE

Et tant d'autres : http://www.security-hacks.com/2007/05/18/top-15-free-sql-injection-scanners

MySql Migration ToolKit et DB Converter

Il permet de récupérer des bases SQL Server, Oracle et Access.
D'autres produits existent : http://www.01net.com/article/303425.html
Sous Windows, vous pouvez utiliser DB Converter.

Performances

MyTop
mysqlidxchk : détection des index inutilisés
Statpack
MySql Proxy http://dev.mysql.com/downloads/mysql-proxy/
Il apour but de rediriger les écritures sur le maître et les lectures sur l'esclave, dans le cas de réplication.

Benchmarking

MySQL Benchmark Suite : fourni dans le code source des dsitributions
MySQL Super-Smack
MyBench

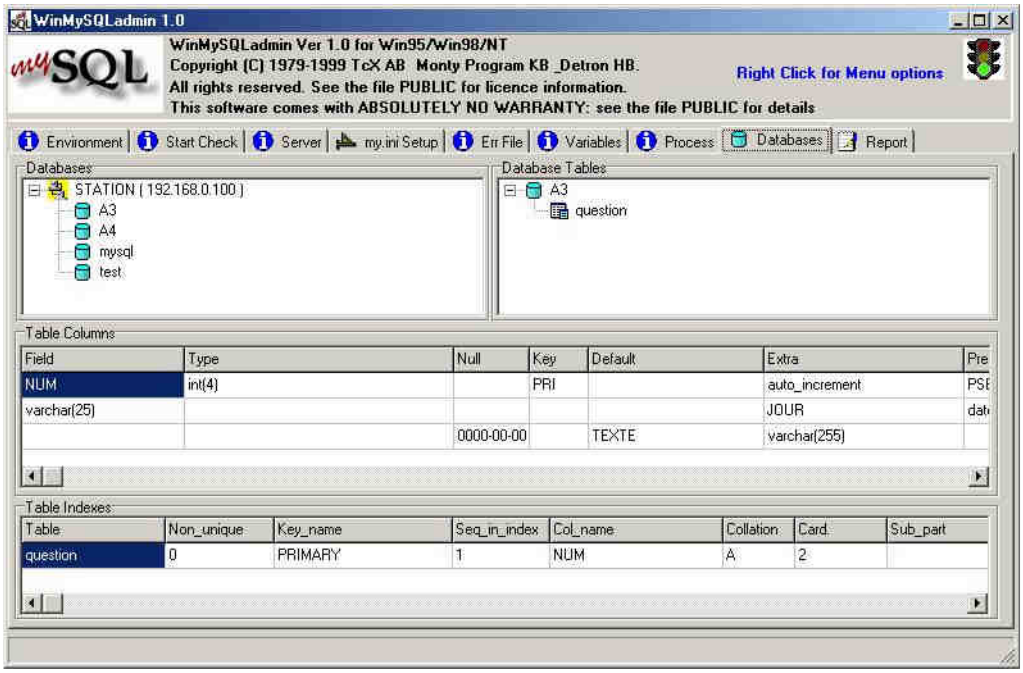
MySql toolkit

Réplication

mysqlbinlog
check_repl
fix_repl
purge_binary_logs
mysqldiff
write_heartbeat
read_heartbeat

WinMySQLAdmin

L'exécutable est stocké par défaut dans c:\MYSQL\BIN sous le nom WINMYSQLADMIN.EXE.



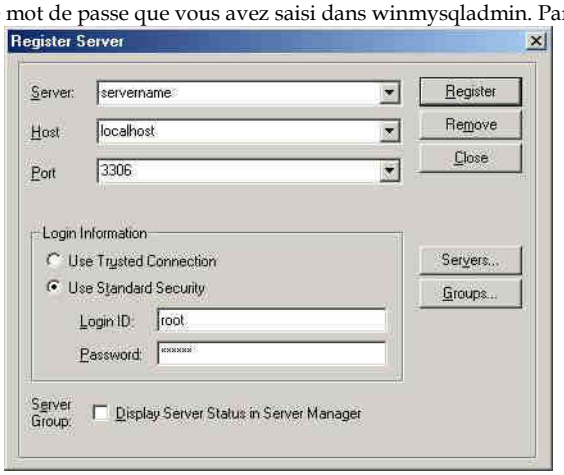
Par un clic droit, vous pouvez créer une base de données.
Lors de la première exécution de WinMySQLAdmin, vous devez entrer le mot de passe superuser. Par défaut, c'est my_password.

Le mot de passe est affiché en clair sous Windows dans le fichier c:\WINDOWS\MY.INI.

MySQLManager

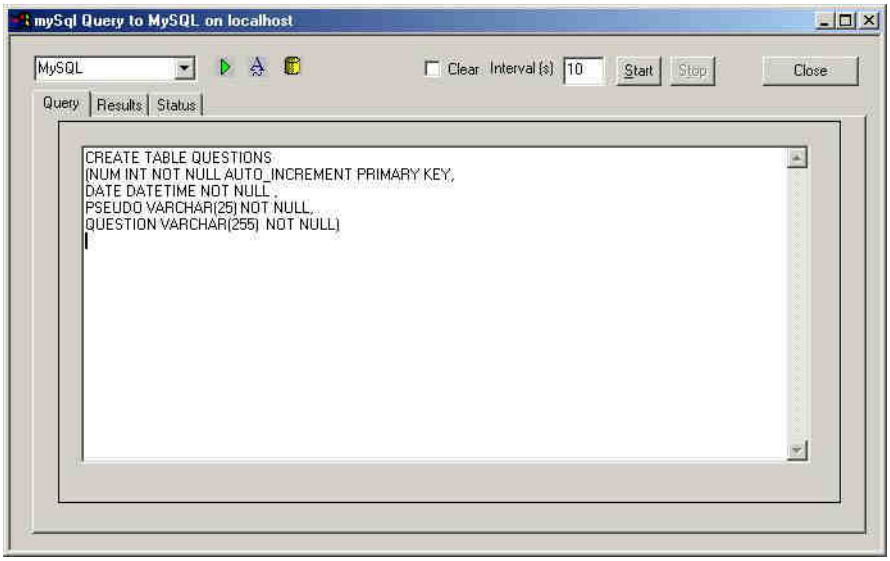
Connexion

A partir de l'explorateur, double-cliquez sur le fichier c:\MYSQL\BIN\MYSQLMANAGER.EXE.
Dans le menu Tools | Register Server, entrez le mot de passe que vous avez saisi dans winmysqladmin. Par défaut, le mot de passe est my_password.



Ordres SQL

Après avoir choisi la base de données préalablement créé avec l'ordre USE, allez dans le menu Tools | SQLQuery.



Création de table

```
CREATE TABLE QUESTIONS
(NUM INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
DATE DATETIME NOT NULL ,
PSEUDO VARCHAR(25) NOT NULL,
QUESTION VARCHAR(255)  NOT NULL)
```

Les types de champ

L'utilisation d'un champ blob engendre une augmentation de la taille des bases de données et des fichiers qui stockent les données. Les performances s'en trouvent affectées. De ce fait, il est préférable de stocker dans la base le lien qui permet d'accéder au champ binaire : image, vidéo,e xécutable.

Intégration des contraintes d'intégrité

Les ocntraintes de typ check ne sont toujours pas possibles dans la version 5.0 de MySql. C'est une absence fonctionnelle fort dommeagable qui oblige les développeurs à gérer les contraintes dans les applications.

```
CREATE TABLE `car_event` (
`internal_number` int(11) NOT NULL COMMENT 'Resp.:Office clerk',
`moment` datetime NOT NULL COMMENT 'Resp.: store assistant',
`event_code` int(11) NOT NULL COMMENT 'Resp.: store assistant',
PRIMARY KEY (`internal_number`),
KEY `event_code` (`event_code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `car_event`
ADD CONSTRAINT `car_event_ibfk_1` FOREIGN KEY (`event_code`) REFERENCES
```

Installation de mytop

Les modules Perl

- DBI
- DBD::mysql
- Term::ReadKey

- Time::HiRes
- Term::ANSIColor

Installation

```
wget http://jeremy.zawodny.com/mysql/mytop/mytop-1.5.tar.gz
tar zxvf mytop-1.5.tar.gz
cd mytop-1.5
perl Makefile.PL
make install
```

Fichier de configuration ~/.mytop

```
user=jzawodn
pass=blah!db
host=localhost
```

Les commandes et utilitaires

Mysql : console texte permettant d'entrer des ordres MySql

Modification du prompt

Syntaxe

mysql –h nom_hôte|adresse_ip –u utilisateur –p mot_de_passe

Commutateurs

-h permet le choix du serveur
-u fixe le nom de l'utilisateur
-p Le commutateur permet de demander le mot de passe

Changer le prompt

export MYSQL_PS1="(\u@\h) [\d]> "
Dans le fichier my.cnf

```
[mysql]

prompt=(\\u@\\h) [\\d]>\\_
Sous l'interpréteur :
prompt (\u@\h) [\d]>\
```

Commandes console

Commande	Raccourci	Signification
Help| ?	\h|\?	Affiche les commandes console
clear	\c	Commande d'effacement
connect	\r	Reconnexion au serveur. Les arguments facultatifs sont db ou host.
ego	\G	Envoie une commande au serveur et affiche le résultat verticalement (?)
Exit|quit	\q	Quitte la console
go	\g	Envoie une commande
notee	\t	N'écrit pas dans un fichier de sortie
print	\p	Imprime la commande courante
rehash	\#	Rebuild completion hash.
source	\.	Exécute un script SQL
status	\s	Informations sur le serveur MySql
tee	\T	Envoie vers un fichier (mouchard)
use	\u	Ouvre une base de données

\v	The server version
\d	The current database
\h	The server host
\p	The current TCP/IP host
\u	Your username
\U	Your full user_name@host_name account name
\\	A literal '\' backslash character
\n	A newline character
\t	A tab character
\	A space (a space follows the backslash)
_	A space
\R	The current time, in 24-hour military time (0-23)
\r	The current time, standard 12-hour time (1-12)
\m	Minutes of the current time
\y	The current year, two digits
\Y	The current year, four digits
\D	The full current date
\s	Seconds of the current time
\w	The current day of the week in three-letter format (Mon, Tue, ...)

\P	am/pm
\o	The current month in numeric format
\O	The current month in three-letter format (Jan, Feb, ...)
\c	A counter that increments for each statement you issue

Exemple

```
D:\mysql\bin>mysql -h 192.168.0.100 -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 3.23.36

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql>
```

Variables d'environnement

Variable	Description
CXX	The name of your C++ compiler (for running configure).
CC	The name of your C compiler (for running configure).
CFLAGS	Flags for your C compiler (for running configure).
CXXFLAGS	Flags for your C++ compiler (for running configure).
DBI_USER	The default username for Perl DBI.
DBI_TRACE	Trace options for Perl DBI.
HOME	The default path for the mysql history file is $HOME/.mysql_history.
LD_RUN_PATH	Used to specify where your libmysqlclient.so is located.
MYSQL_DEBUG	Debug trace options when debugging.
MYSQL_HISTFILE	The path to the mysql history file. If this variable is set, its value overrides the default of $HOME/.mysql_history.
MYSQL_HOST	The default hostname used by the mysql command-line client.
MYSQL_PS1	The command prompt to use in the mysql command-line client.
MYSQL_PWD	The default password when connecting to mysqld. Note that use of this is insecure! See Section 4.5.6, "Keeping Your Password Secure."
MYSQL_TCP_PORT	The default TCP/IP port number.
MYSQL_UNIX_PORT	The default Unix socket filename; used for connections to localhost.
PATH	Used by the shell to find MySQL programs.
TMPDIR	The directory where temporary files are created.
TZ	This should be set to your local time zone. See Section A.4.6, "Time Zone Problems."
UMASK_DIR	The user-directory creation mask when creating directories. Note that this is ANDed with UMASK!
UMASK	The user-file creation mask when creating files
USER	The default username on Windows and NetWare to use when connecting to mysqld.

mysql_install_db

Ce script permet d'installer les fichiers de base liés au fonctionnement de MySql. Cette commande doit être souvent employé après uen désinstallation et une réinstallation du paquet MySql.

mysqlc

Identique en tout point à mysql. Exige la présence de cygwin.dll dans le répertoire %WINDIR%\SYSTEM32 pour NT/XP/2000 et %WINDIR%\SYSTEM pour 9x.

mysqlshow : visualise les bases et leurs contenus (tables)

Syntaxe

mysqlshow [--keys|--status] base_de_données table
Affiche les infos de la base de données ou celle de la table

Commutateurs

--keys affiche les informations sur les index

--status affiche les infos concernant les tables de la base de données

Exemple

Lister toutes les bases de données : mysqlshow
Lister toutes les tables contenues dans une base de données : mysqlshow DSFC

mysqldump

Syntaxe

mysqldump [--no-data|--extended-insert] base_de_données table

Commutateurs

--no-data Affiche les informations concernant la table
--extended-insert Permet la sauvegarde de la base de données

Exemple

Sauvegarde les instructions SQL dans une tables : mysqldump DSFC >script.sql

Mysqladmin : permet l'exécution de commandes d'administration à partir du shell

Syntaxe

mysqladmin [OPTIONS] commande1 command2…

Les options

-h, --host=# Spécifie la machine à laquelle se connecter
-p, --password[=...] Permet de spécifier le mot de passe
-P --port=... Spécifie le port
-u, --user=# Spécifie l'utilisateur

Les options par défaut sont lues sous Windows à partir des fichiers D:\WINDOWS\my.ini C:\my.cnf .

Les commandes

create databasename	Création d'une nouvelle base de données
drop databasename	Effacement d'une base de données
flush-hosts	Actualise les machines
flush-logs	Actualise les logs
flush-status	Efface les variables
flush-tables	Actualise les tables
flush-threads	Actualise le cache
flush-privileges|reload	Recharge la table des privilèges
kill id,id,...	Tue les threads MySql
password new-password	Change le mot de passe
ping	Test la présence du démon MySql
processlist	Liste les threads actifs au niveau du serveur
refresh	Recharge toutes les tables
shutdown	Arrête le serveur
status	Affiche un message sur le fonctionnement du serveur
start-slave	??? (SM)
stop-slave	??? (SM)
variables	Affiche les variables
version	Affiche l'information sur la version du serveur

Exemples

msqladmin variables
mysqladmin -u status
mysqladmin -u root reload
mysladmin shutdown

Mysqlimport : importation de données dans une table existante

Cette commande est identique à la commande SQL LOAD DATA INFILE.

Redirection

Le symbole de redirection

```
D:\mysql\bin>mysql -h 192.168.0.100 -u root < script.sql
```

Contenu de script.sql

show databases;
use A3;
describe question;

Mysqlwatch

mysqlwatch –install	Installe le service
mysqlwatch –remove	Retire le service

Perror

Affiche le numéro de l'erreur générée par MySql

Myisampack ou pack_isam (ancien format): commande de compression des tables

-b, --backup	Fait une sauvegarde de la table d'origine dans nom_table.OLD (vivement conseillé)
-f, --force	Pas d'affichage d'erreur si le fichier temporaire nom_table.TMD est toujours présent
-j big_tbl_name, --join=big_tbl_name	Assemble toutes les tables spécifiées au niveau de la ligne de commande. Le définition des tables doit être rigoureusement identique
-s, --silent	Mode silencieux
-t, --test	Effectue un simple Test
-T dir_name, --tmp_dir=dir_name	Spécifie le répertoire temporaire
-v, --verbose	Mode verbeux (type discours Jacques Chirac ou Edouard Balladur)
-V, --version	Information de version
-w, --wait	

myisamchk ou isamchk (pour les anciens formats)

Test

Pensez à verrouiller les tables avant de vérifier par :
LOCK nom_table READ
myisamchk nom_de_table vérifie les tables

Réparation

Pensez à verrouiller les tables avant de réparer par :
myisamchk --recover | --safe-recover répare les tables
Après avoir réparé, videz le cache de la table et libérez les verrous :
FLUSH TABLES.
UNLOCK TABLES;

Mysqlcheck : répare et optimise les tables

Les options

-c,-m,-C	diagnostique
-r	répare les erreurs
-a	analyse les erreurs
-o	optimise les tables

Syntaxe

mysqlcheck [OPTIONS] database [tables]
mysqlcheck [OPTIONS] --databases DB1 [DB2 DB3...]
mysqlcheck [OPTIONS] --all-databases

mysqlbinlog

Pour la réplication

Commandes MySQL

Bases de données

Commandes

CREATE DATABASE nom_base_de_données;	Crée une base de données (64 caractères maximum).
USE nom_base_de_données;	sélectionne une base de données
SELECT DATABASE();	visualise la base de données en cours
SHOW DATABASES;	montre toutes les bases de données
DROP nom_base_de_données;	détruit la base de données

Exemple

```
mysql> create database A4;
Query OK, 1 row affected (0.06 sec)

mysql> use A4;
Database changed

mysql> select database();
+------------+
| database() |
```

```
+------------+
| A4         |
+------------+
1 row in set (0.06 sec)

mysql> show databases;
+----------+
| Database |
+----------+
| A3       |
| A4       |
| mysql    |
| test     |
+----------+
4 rows in set (0.00 sec)

mysql>
```

Informations sur les tables

Commandes

SHOW TABLES;
DESCRIBE nom_de_table;

Exemples

```
mysql> show tables;
+--------------+
| Tables_in_A3 |
+--------------+
| question     |
+--------------+
1 row in set (0.00 sec)

mysql> describe question;
+--------+--------------+------+-----+------------+----------------+
| Field  | Type         | Null | Key | Default    | Extra          |
+--------+--------------+------+-----+------------+----------------+
| NUM    | int(4)       |      | PRI | NULL       | auto_increment |
| PSEUDO | varchar(25)  |      |     |            |                |
| JOUR   | date         |      |     | 0000-00-00 |                |
| TEXTE  | varchar(255) |      |     |            |                |
+--------+--------------+------+-----+------------+----------------+
4 rows in set (0.05 sec)

mysql>
```

Manipulation des tables

Création de table

CREATE TABLE Nom_De_Table(
Champ INT(N) | FLOAT(N,D) | DOUBLE(N,D) | DATETIME
CHAR(N) | VARCHAR(N) NOT NULL DEFAULT Valeur_Par_Défaut AUTO_INCREMENT PRIMARY KEY
);

Défragmentation

OPTIMIZE TABLE Nom_De_Table

Suppression

DROP TABLE Nom_De_Table

Modification

ALTER TABLE Nom_De_Table

la commande MERGE sur tables ISAM

```
CREATE TABLE mytable (

  a    INTEGER  NOT NULL PRIMARY KEY,

  b    CHAR(18) NOT NULL
```

) TYPE = MERGE UNION = (mytable0, mytable1, mytable2) INSERT_METHOD = LAST | NO | FIRST;

Manipulations des données

Ajout

INSERT INTO Nom_De_Table (Champ1,…, ChampN) VALUES (Valeur1,…, Valeur N)

Modification

UPDATE Nom_De_Table SET Champ1= Valeur1,…, ChampN= ValeurN WHERE…

Suppression

DELETE FROM Nom_De_Table WHERE…

Instruction SELECT

query_cache=1
SELECT SQL_NO_CACHE * FROM mytable
SELECT SQL_CACHE * FROM mytable
SELECT NOW(),USER(),VERSION();
SELECT COUNT(*|[DISTINCT] champ) [AS alias] FROM table;
Préférez IN pour des raisons de lisibilité. Du point de vue performance, MySQL transcrit le IN en OR.
SELECT * FROM mytable USE INDEX (mod_time, name) ...
SELECT * FROM mytale IGNORE INDEX (priority) ...
SELECT * FROM mytable FORCE INDEX (mod_time) ...

Mode de fonctionnement en lecture seule

```
mysql -h localhost -u root -p
mysql> set GLOBAL read_only=true;
mysql> set GLOBAL read_only=false;
mysql> FLUSH TABLES WITH READ LOCK;
mysql> UNLOCK TABLES;
mysql> SHOW PROCESSLIST;
| 5 | root | localhost | test | Query | 160 | Waiting for release of readlock | INSERT INTO toto VALUE('tata2') |
```

Fonctions

LAST_INSERT_ID(champ)

Verrous

LOCK TABLES table1 WRITE, table2, READ, …;
UNLOCK TABLES

Requêtes complexes

Jointures

Nom	Définition	Exemple
Cross join	Jointure croisée	Renvoie toutes les combinaisons possibles. SELECT * FROM table1 CROSS JOIN table2;
Inner join	Jointure interne	C'est la jointure par défaut. Les lignes renvoyées correspondent aux intersections entre les tables. Pour MySQL, jointure croisée et jointure interne sont équivalentes, ce qui n'est pas le cas en SQL standard.
Left outer join	Jointure externe gauche	Limite les résultats à ceux incluant chaque ligne de la table à gauche du "JOIN", ici table1. SELECT * FROM table1 LEFT JOIN table2;
Right outer join	Jointure externe droite	Limite les résultats à ceux incluant chaque ligne de la table à droite du "JOIN", ici table2. SELECT * FROM table1 RIGHT JOIN table2;
Full outer join	Jointure externe complète	Toutes les lignes des deux tables sont incluses, et les lignes n'ayant pas de correspondances avec celles de l'autre table sont associées à des NULL. MySQL ne reconnaît pas les jointures externes complètes, mais cela peut être compensé par l'utilisation d'une union.

Unions

Les unions ont un fonctionnement proche des jointures : combiner des informations en provenance de plusieurs sources, en l'occurrence ici des requêtes.
La différence réside dans le fait que les données combinées doivent être du même type, et les tables doivent avoir les mêmes noms de colonnes. Les doublons sont éliminés, sauf si l'on précise le mot-clef ALL, auquel cas les valeurs multiples sont renvoyées autant de fois qu'elles apparaissent dans toutes les tables.
Pour les mettre en place, les parenthèses sont obligatoires pour chaque requête.

```
(SELECT nom FROM employes_france)
UNION (SELECT nom FROM employes_espagne)
UNION (SELECT nom FROM employes_italie);
```

Sous-requêtes

Une sous-requête consiste à établir une requête SQL traitant les données issues d'une seconde requête. La clause WHERE comprend ainsi une autre requête SQL pour limiter les résultats originels, ou pour agréger les informations de plusieurs tables. Une sous-requête peut également être placée dans une clause HAVING.

```
SELECT * FROM table WHERE colonne1 = (SELECT colonne2 FROM table2);
SELECT * FROM table WHERE colonne = (SELECT colonne2 FROM table2 WHERE colonne2 = (SELECT colonne3 FROM table3));
```

Table temporaire

L'utilisation d'une table temporaire ne dépasse jamais la session SQL en cours, et autorise la création de tables pour créer un nouvel arrangement d'informations plus adapté à une requête particulière, pour tester la bonne insertion de données dans la base, ou pour créer un sous-ensemble d'une autre table plus léger afin de soulager le serveur SQL... On ne peut logiquement y faire appel qu'une seule fois dans la même requête.

```
CREATE TEMPORARY TABLE temp SELECT * FROM donnes WHERE a=1;
SELECT * FROM temp WHERE ... ;
```

Transactions

Begin

```
BEGIN; COMMIT;
BEGIN; ROLLBACK;
```

Autocommit

```
set AUTOCOMMIT = 1|0;
```

Cohérence

```
SET GLOBAL|SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED | READ UNCOMMITTED | REPEATABLE READ | SERIALIZABLE
```

MyIsam ou InnoDb

MyIsam

L'index de type Full Index Text sur les champs TEXT ou BLOB n'est possible uniquement avec MyIsam. Il s'agit d'un mode qui indexe tous les mots d'un champ.
Même s'il n'existe pas de limite en dehors de celle des systèmes de fichiers associés Aux systèmes d'exploitation, il est conseillé de limiter la taille des ables à 4GB.
La gestion des verrous se fait au niveau table.
Pas de possibilité d'avoir de clusters sachant que chaque index est dans un fichier séparé

InnoDb

Le Mode transactionnel est supporté uniquement par le moteur InnoDb.
Les index sont regroupés aux côtés de la clé primaire.

Gestion du cache et des index

Alors que InnoDb gère un cache sur données avec innodb_buffer_pool , le système MyISAM s'appuie sur le système d'exploitation. Le paramètre key_buffer permet le stockage des blocs d'index.

Les autres Moteurs

Heap

Les données sont en mémoire alors que la définition de table est sur disque.

ndb ou Network DataBase

Il s'agit du mode cluster.

Comparatif

Attribute	MyISAM	Heap	BDB	InnoDB
Transactions	No	No	Yes	Yes
Lock granularity	Table	Table	Page (8 KB)	Row
Storage	Split files	In-memory	Single file per table	Tablespace(s)
Isolation levels	None	None	Read committed	All

Portable format	Yes	N/A	No	Yes
Referential integrity	No	No	No	Yes
Primary key with data	No	No	Yes	Yes
MySQL caches data records	No	Yes	Yes	Yes
Availability	All versions	All versions	MySQL-Max	All Versions

Choix du moteur

```
ALTER TABLE mytable engine = InnoDB;
BEGIN;

INSERT INTO innodb_table SELECT * FROM myisam_table WHERE id BETWEEN x AND y;

COMMIT;

CREATE TABLE newtable LIKE mytable;

INSERT INTO newtable SELECT * FROM mytable;
```

A creuser

```
GROUP_CONCAT(name)
explain \G         Analyseur de requêtes²
```

La sécurité

Système de cryptage

MySQL utilise son propre système de cryptage que vous pouvez activer avec la fonction PASSWORD. Dans la table MySQL.user, le mot de passe est crypté selon la méthode employée par la fonction PASSWORD().

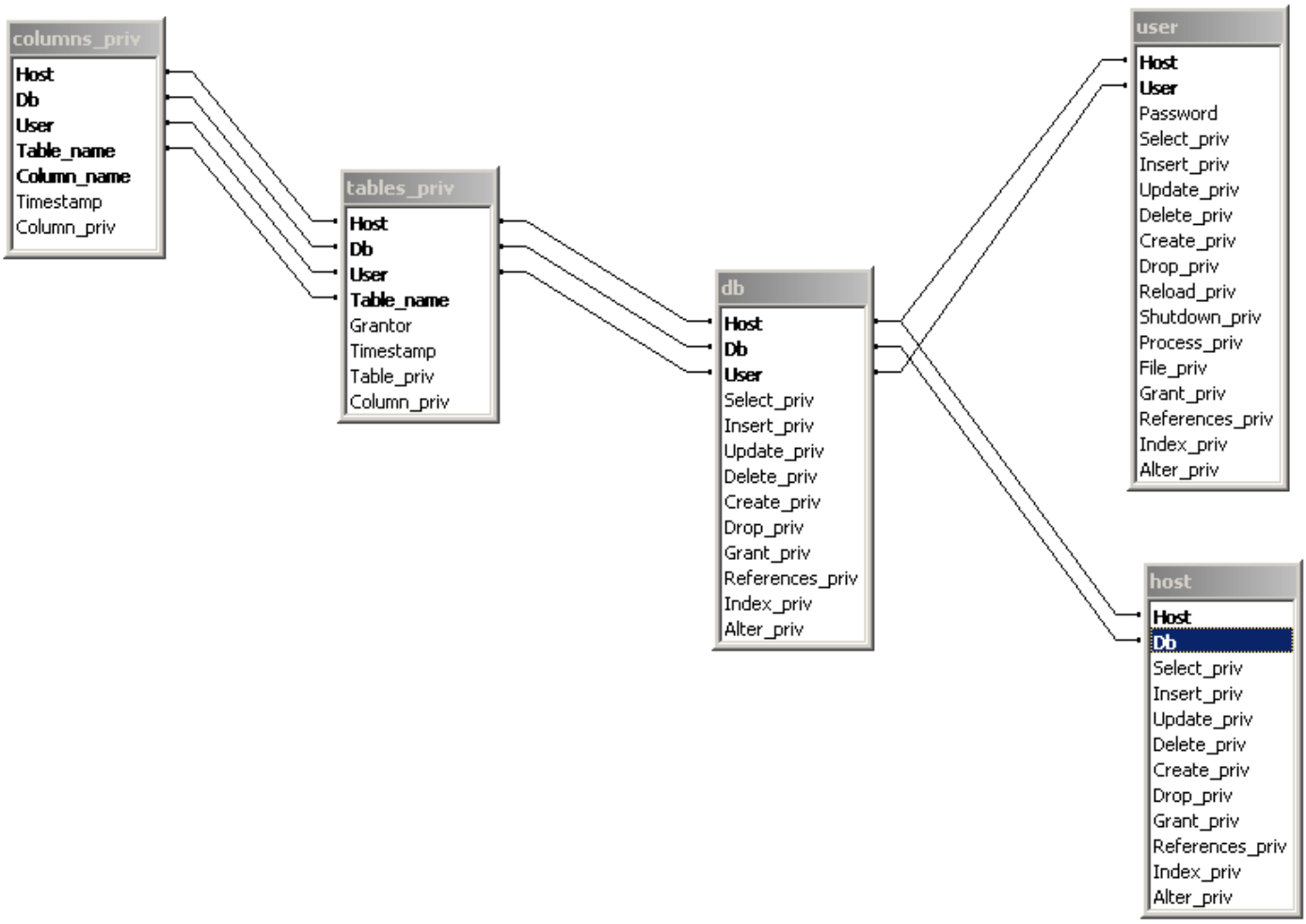
Changer le mot de passe root

mysql –u root
Entrez l'ancien mot de passe.
Pour changer le mot de passe root, vous pouvez employer la commande SQL :
UPDATE user SET Password=PASSWORD("Votre_nouveau_mot_de_passe") WHERE user='root';
SET PASSWORD FOR root=PASSWORD("votre_nouveau_mot_de_passe")
La commande FLUSH PRIVILEGES permet de vider le cache contenant les informations lues précédemment à partir des tables. Elle est nécessaire si vous intervenez sur table.

Créer un nouveau mot de passe pour l'administrateur MySQL

service mysqld stop
mysqld_safe --skip-grant-tables -u root &
mysqladmin -u root flush-privileges \ password "MonNouveauMotDePasse"
service mysqld restart

Le schéma des tables liées à la gestion de droits



La table user

Elle contient les utilisateurs et leurs privilèges sur la table elle-même.

Structure

```
Host
User
Password
Select_priv
Insert_priv
Update_priv
Delete_priv
Create_priv
Drop_priv
Reload_priv
Shutdown_priv
Process_priv
File_priv
Grant_priv
References_priv
Index_priv
Alter_priv
```

Ajout

INSERT INTO MySQL.user(host,user,password)
VALUES('192.168.0.100','root',PASSWORD("c10evs"))

La table host

Elle contient les machines autorisées à réaliser certaines opérations.

```
Host
Db
Select_priv
Insert_priv
Update_priv
Delete_priv
Create_priv
Drop_priv
Grant_priv
```

```
References_priv
Index_priv
Alter_priv
```

La table db

Elle contient les niveaux de privilège sur la base de données.

```
Host
Db
User
Select_priv
Insert_priv
Update_priv
Delete_priv
Create_priv
Drop_priv
Grant_priv
References_priv
Index_priv
Alter_priv
```

La table tables_priv

Elle contient les niveaux de privilèges sur les tables.

```
Host
Db
User
Table_name
Grantor
Timestamp
Table_priv
Column_priv
```

La table columns_prv

Elle Contient les niveaux de privilèges sur les colonnes.

```
Host
Db
User
Table_name
Column_name
Timestamp
Column_priv
```

Remarques sur les modifications de droit

Préférez l'emploi des Commandes GRANT et REVOKE qui modifient respectivement les tables DB, user, tables_priv, columns_priv.
Pour que soient prises en compte les modifications, exécutez la commande FLUSH_PRIVILEGES.
La casse n'est pas prise en compte sur les noms de machines, les champs.

GRANT

Définition

Cette commande SQL permet de donner les droits sur les tables et les champs des tables.

Syntaxe

GRANT ALTER|CREATE|DELETE|DROP|INDEX|INSERT|REFERENCES|SELECT|UPDATE
|FILE|PROCESS|RELOAD|SHUTDOWN
|USAGE|ALL
ON Base_de_Données.*|*.*|Base_de_Données.Table
TO Utilisateur
IDENDIFIED BY "Mot_De_Passe"
WITH GRANT OPTION
La clause WITH GRANT OPTION permet de donner des droits à gérer des droits

Exemple

GRANT SELECT ON *.* to 'root@'localhost' IDENTIFIED BY 'Pass';

REVOKE

REVOKE
GRANT ALTER|CREATE|DELETE|DROP|INDEX|INSERT|REFERENCES|SELECT|UPDATE
|FILE|PROCESS|RELOAD|SHUTDOWN
|USAGE|ALL
ON Base_de_Données.*|*.*|Base_de_Données.Table
FROM Utilisateur

Exemple

GRANT ALL ON *.* TO root@localhost IDENTIFIED BY "c10evs";

La commande suivante crée l'utilisateur et donne des droits SELECT sur la table A3 de la base A3 au niveau de la machine localhost. Les tables user,
GRANT SELECT ON A3.questions TO Pierre@localhost IDENTIFIED BY "PLF";
REVOKE ALL ON *.* FROM Pierre;
GRANT SELECT (user) ON MySQL.user TO Pierre@localhost IDENTIFIED BY "PLF";

Script de test Php (Apache doit être installé sur le poste)

```
<?
    $CONNECT_SERVER=@mysql_connect("localhost","Pierre","PLF") or die("Connexion impossible !");
    echo "Connexion ok<BR>";
    @mysql_query("SELECT * FROM MySQL.user",$CONNECT_SERVER) or die("Requête interdite");
    echo "Requête ok";
?>
```

Crypter les champs

Mot de passe

```
INSERT INTO utilisateur(id_utilisateur, mdp_utilisateur) VALUES (50 MD5('choucroute'));
```

Contenu sécurisé

```
INSERT INTO document(titre, contenu) VALUES ('dur dur', AES_ENCRYPT('gazou','mdp'));
SELECT titre, AES_DECRYPT(contenu,'mdp') AS contenu FROM document;
```

Les méthodes de cryptage

MD5('mot de passe')
PASSWORD(...)
ENCRYPT(...)

Privilèges globaux

Etendue

Reload	FLUSH TABLES, FLUSH STATUS
Shutdown	Arrêt du serveur
Process	SHOW PROCESSLIST, KILL
File	LOAD DATA INFILE
Super	Possibilité de tuer n'importe quelle requête

Exemple

```
GRANT PROCESS, SHUTDOWN on *.*
TO 'root'@'localhost'
IDENTIFIED BY 'root';
```

Récapitulatif

Privilege	Column	Context
ALTER	Alter_priv	tables
DELETE	Delete_priv	tables
INDEX	Index_priv	tables
INSERT	Insert_priv	tables
SELECT	Select_priv	tables
UPDATE	Update_priv	tables
CREATE	Create_priv	databases, tables, or indexes
DROP	Drop_priv	databases or tables
GRANT	Grant_priv	databases or tables
REFERENCES	References_priv	databases or tables
CREATE TEMPORARY TABLES	Create_tmp_table_priv	server administration
EXECUTE	Execute_priv	server administration
FILE	File_priv	file access on server host
LOCK TABLES	Lock_tables_priv	server administration
PROCESS	Process_priv	server administration
RELOAD	Reload_priv	server administration
REPLICATION CLIENT	Repl_client_priv	server administration
REPLICATION SLAVE	Repl_slave_priv	server administration
SHOW DATABASES	Show_db_priv	server administration
SHUTDOWN	Shutdown_priv	server administration
SUPER	Super_priv	server administration
USAGE		Compte sans privilèges

Limitation

Contrairement à Oracle, MySql ne possède ni rôle (groupe d'utilisateurs disposant de privilèges), ni profils (restrictions de consommation de ressources)

La réplication

Créationdes comptes sur le client et le serveur

```
GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO

repl@"192.168.1.0/255.255.255.0" IDENTIFIED BY 'c0pyIT!';

SHOW GRANTS FOR repl;
```

my.cnf au niveau master

```
log-bin

server-id = 1
```

my.cnf au niveau slave

server-id = 2

master-host = master.example.com

master-user = repl

master-password = c0pyIT!

master-port = 3306

START SLAVE;

Redémarrage

Tout d'abord, redémarrez le maître. Puis redémarrez l'esclave.

Réplication des données

FLUSH TABLES WITH READ LOCK (Isam)
FLUSH MASTER

ou bien

LOAD TABLE mytable FROM MASTER
LOAD DATA FROM MASTER : la base mysql n'est pas sauvegardée.

Filtrage à partir du maître

Sur la base

```
binlog-do-db=dbname
```
binlog-ignore-db=dbname

Sur la table

```
replicate-do-table=dbname.tablename

replicate-ignore-table=dbname.tablename

replicate-wild-do-table=dbname.tablename

replicate-wild-ignore-table=dbname.tablename

replicate-do-db=dbname

replicate-ignore-db=dbname

replicate-rewrite-db=from_dbname->to_dbname
```

Etat de la réplication

```
SHOW MASTER STATUS \G
SHOW MASTER LOGS;
SHOW SLAVE STATUS \G
```

Changer de maître

Côté du nouveau maître

RESET MASTER

Côté des anciens esclaves

```
CHANGE MASTER TO
    -> MASTER_HOST='newmaster.example.com',
    -> MASTER_USER='repl',
    -> MASTER_PASSWORD='MySecret!',
    -> MASTER_PORT=3306,
    -> MASTER_LOG_FILE='log-bin.001',
    -> MASTER_LOG_POS=4;
```

Purge des logs

PURGE MASTER LOGS TO 'binary-log.004';

Encodage

Passer en UTF-8

```
ALTER TABLE `table_name`  DEFAULT CHARACTER SET utf8 COLLATE utf8_bin
```

Les variables

Performances

key_buffer_size stocke en cache les blocks d'index en mode Isam. Vous pouvez prendre jusquà 30 à 40% de la mémoire disponible.

innodb_buffer_pool_size Pour le moteur InnoDb, gère le cache des données et des index (jusqu'à 70 à 80% de la mémoire du système)

innodb_additional_mem_pool_size 20 MB

innodb_log_file_size entre 64MB et 512MB

innodb_log_buffer_size entre 8MB et 16MB

innodb_flush_logs_at_trx_commit Entre 0 et 2
O est plus rapide et moins sûre.

table_cache 1024 pour 200 tables

thread_cache 16

query_cache_size entre 32M et 512M

http://dev.mysql.com/doc/refman/5.0/fr/server-system-variables.html

back_log	50
basedir	d:\mysql\
binlog_cache_size	32768
character_set	latin1
character_sets	latin1 big5 czech euc_kr gb2312 gbk sjis tis620 ujis dec8 dos german1 hp8 koi8_ru latin2 swe7 usa7 cp1251 danish hebrew win1251 estonia hungarian koi8_ukr win1251ukr greek win1250 croat cp1257 latin5
concurrent_insert	ON
connect_timeout	5
datadir	f:\appli\data\
delay_key_write	ON
delayed_insert_limit	100
delayed_insert_timeout	300
delayed_queue_size	1000
flush	OFF
flush_time	1800
have_bdb	NO
have_gemini	NO
have_innodb	NO
have_isam	YES
have_raid	NO
have_openssl	NO
init_file	

interactive_timeout	28800
join_buffer_size	131072
key_buffer_size	16773120
language	d:\mysql\share\english\
large_files_support	ON
log	OFF
log_update	OFF
log_bin	OFF
log_slave_updates	OFF
log_long_queries	OFF
long_query_time	10
low_priority_updates	OFF
lower_case_table_names	1
max_allowed_packet	1047552
max_binlog_cache_size	4294967295
max_binlog_size	1073741824
max_connections	100
max_connect_errors	10
max_delayed_threads	20
max_heap_table_size	16777216
max_join_size	4294967295
max_sort_length	1024
max_user_connections	0
max_tmp_tables	32
max_write_lock_count	4294967295
myisam_max_extra_sort_file_size	256
myisam_max_sort_file_size	2047
myisam_recover_options	0
myisam_sort_buffer_size	8388608
net_buffer_length	16384
net_read_timeout	30
net_retry_count	10
net_write_timeout	60
open_files_limit	0
pid_file	f:\appli\data\station.pid
port	3306
protocol_version	10
record_buffer	131072
record_rnd_buffer	131072
query_buffer_size	0
safe_show_database	OFF
server_id	0
slave_net_timeout	3600
skip_locking	ON
skip_networking	OFF
skip_show_database	OFF
slow_launch_time	2
socket	MySQL
sort_buffer	2097144
sql_mode	0
table_cache	64
table_type	MYISAM
thread_cache_size	0
thread_stack	131072
transaction_isolation	READ-COMMITTED
timezone	Paris, Madrid
tmp_table_size	33554432
tmpdir	D:\WINDOWS\TEMP\
version	3.23.45-nt
wait_timeout	28800

Tableau: instructions de procédures stockées supportées dans MySQL 5.0

Instruction	Description
CREATE PROCEDURE	Crée une procédure stockée qui est stockée dans la table proc de la base de données MySQL.
CREATE FUNCTION	Crée une fonction définie par l'utilisateur, essentiellement une procédure stockée qui renvoie des données.
ALTER PROCEDURE	Modifie une procédure stockée précédemment définie, qui a été créée à l'aide de l'instruction CREATE PROCEDURE. Ne modifie pas les procédures stockées ni les fonctions stockées associées.
ALTER FUNCTION	Modifie une fonction stockée précédemment définie, qui a été créée à l'aide de l'instruction CREATE FUNCTION. Ne modifie pas les procédures stockées ni les fonctions stockées associées.
DROP PROCEDURE	Supprime une ou plusieurs procédures stockées de la table proc de MySQL.
DROP FUNCTION	Supprime une ou plusieurs fonctions stockées de la table proc de MySQL.
SHOW CREATE PROCEDURE	Renvoie le texte d'une procédure stockée précédemment définie, qui a été créée à l'aide de l'instruction CREATE PROCEDURE. Cette instruction est une extension MySQL de la spécification SQL:2003.
SHOW CREATE FUNCTION	Renvoie le texte d'une fonction stockée précédemment définie, qui a été créée à l'aide de l'instruction CREATE FUNCTION. Cette instruction est une extension MySQL de la spécification SQL:2003.
SHOW PROCEDURE STATUS	Renvoie les caractéristiques d'une procédure stockée précédemment définie, notamment son nom, son type, son créateur, sa date de création et sa date de modification. Cette instruction est une extension MySQL de la spécification SQL:2003.
SHOW FUNCTION STATUS	Renvoie les caractéristiques d'une fonction stockée précédemment définie, notamment son nom, son type, son créateur, sa date de création et sa date de modification. Cette instruction est une extension MySQL de la spécification SQL:2003.
CALL	Appelle une procédure stockée précédemment définie, qui a été créée à l'aide de l'instruction CREATE PROCEDURE.
BEGIN ... END	Contient un groupe de plusieurs instructions en vue de leur exécution.
DECLARE	Sert à définir des variables locales, des conditions, des routines de gestion et des curseurs.
SET	Sert à modifier les valeurs des variables locales et des variables serveur globales.
SELECT ... INTO	Sert à stocker les colonnes indiquées directement dans des variables.
OPEN	Sert à ouvrir un curseur.
FETCH	Extrait la ligne suivante à l'aide du curseur spécifié et avance le curseur d'une ligne.
CLOSE	Sert à fermer un curseur ouvert.
IF	Instruction conditionnelle IF-THEN-ELSE-END IF.
CASE ... WHEN	Construction conditionnelle d'une instruction CASE.
LOOP	Structure de boucle simple; la sortie se fait à l'aide de l'instruction LEAVE.
LEAVE	Sert à sortir des instructions IF, CASE, LOOP, REPEAT et WHILE.
ITERATE	Utilisée dans les boucles pour recommencer au début de la boucle.

REPEAT	Boucle avec le test conditionnel à la fin.
WHILE	Boucle avec le test conditionnel au début.
RETURNS	Renvoie une valeur d'une fonction stockée.

Cas d'utilisation des triggers

You need the contents of your index to differ from the data you are indexing. For example, you're indexing email addresses, and you want to be able to search by domain name. You can't use a regular index for this, because you would need to to a query like:

```
SELECT ... FROM users WHERE email LIKE '%.com'
```

That query would, of course, not use indexes.
The Real Solution

MySQL needs its own built-in calculated indexes. The ability to create an index based on some expression, in our example, on REVERSE(email), and have MySQL automatically figure out how to use it. Maybe some day that will exist in MySQL, but it doesn't currently.
The Temporary Solution for MySQL 5.0

You can use Triggers in MySQL 5.0 to automatically maintain another column, which is then indexed! You'll want to create your table like so:

```
CREATE TABLE users (
  id       INT      NOT NULL auto_increment,
  name     CHAR(50)  NOT NULL,
  email    CHAR(120) NOT NULL,
  r_email  CHAR(120) NOT NULL,
  PRIMARY KEY (id),
  INDEX (email),
  INDEX (r_email)
);
```

Notice that I added an extra column, r_email to store the reversed email address. You can then create a couple of triggers to keep the reversed email address up to date automatically:

```
CREATE TRIGGER users_r_email_in
  BEFORE INSERT ON users
  FOR EACH ROW
    SET NEW.r_email = REVERSE(NEW.email);

CREATE TRIGGER users_r_email_up
  BEFORE UPDATE ON users
  FOR EACH ROW
    SET NEW.r_email = REVERSE(NEW.email);
```

Ta da! Whenever you need to search by domain, just run your query like so:

```
SELECT ... FROM users WHERE r_email LIKE 'moc.%'
```

Or, you could go for the minimal amount of modification of your client code, and use this query instead:

```
SELECT ... FROM users WHERE r_email LIKE REVERSE('%.com')
```

Indexes will be used!

Récupération après Crash

MySQL Crash Recovery

MySQL is known for its stability but as any other application it has bugs so it may crash sometime. Also operation system may be flawed, hardware has problems or simply power can go down which all mean similar things - MySQL Shutdown is unexpected and there could be various inconsistences. And this is not only problem as we'll see.

MySQL has angel process mysqld_safe which will restart MySQL Server in most cases. It is great, unless you have run into some bug which causes it to crash again - such crashes qucikly following one another are kind of worse because they explore many less tested code paths in MySQL and so problem potential is larger.

So lets look at the problem which happen during the crash which might need to take care of or which may seriously affect MySQL Performance.

MyISAM Corruption - If you're writting to MyISAM tables there is very large chance of them becoming corrupted during the crash. Note corruption may be hidden and do not expose itself instantly - you may notice wrong query results days after crash. Sometimes corrupted tables may be reason for further crashes or hangs, and corruption may spread itself further in the table. You probably do not want any of these so it is very good idea to run MySQL with myisam_recover option which will make sure all improperly closed MyISAM tables are checked first time it is accessed. This option is however rather painful to use with web applications - users may issue different queries which may trigger check/repair running for many tables at onces, which typically make system extremely slow and also can use up all allowed connections or run out of memory (myisam_sort_buffer_size is normally set pretty lage). If this becomes the problem I use tiny script which moves out all MyISAM tables out of MySQL database directory, checks them with MyISAMchk and moves them back to running server. This looks scary but it works great - until table is checked and ready application gets error rather than stalling forever which allows application to become partially functional as soon as possible. This hack is needed only in some cases - in most cases using Innodb for tables which you need to be recovered fast is better solution.

Innodb Recovery - Unless you have some hardware problems (99%) or found new Innodb bug (1%) Innodb recovery should be automatic and bring your database to consistent state. Depending on innodb_flush_lot_at_trx_commit setting you may lose few last committed transactions but it is it. It is Performance of this process which may cause the problems. As I already wrote innodb_log_file_size and innodb_buffer_pool_size affect recovery time significantly as well as your workload. I should also mention if you have innodb_file_per_table=1 your recovery speed will depend on number of Innodb tables you have, as well as many other operations, so beware.

Binary log corruption - Binary log may become corrupted and out of sync with database content. This will sometimes break replication but if you're just planning on using binary log for point in time recovery it can go unnoticed. sync_binlog Is helping by syncing binary log, but at performance penalty. If using Innodb you also might with to use innodb-safe-binlog option in MySQL 4.1 so your Innodb log and binary log are synchronized. In MySQL 5.0 XA is taking care of this synchronization.

.frm Corruption - Few people know MySQL is not really ACID even with Innodb tables, at least not for DDL statements.
There is a chance of failing for example during CREATE statement with table created in Innodb dictionary but .frm not created or not completely written. Partially written .frm files or .frm being unsync with internal Innodb dictionary may cause MySQL to fail with wierd error messages. In MySQL 4.1 sync_frm option was added which reduces this problem as time window when it can happen is much less. Still if failure happens just during writting .frm file nasty things may happen, not to mention such potentially multiple operation DDL statements as RENAME TABLE - these are most vulnerable.

master.info corruption - If slave happens to crash you can also have relay logs corruption and master.info being corrupted. Not to mention MyISAM tables can contain partially completed statements as well as some of updates totally lost. The safe approach it to reclone the slaves if they crash or you can take the risks and try to continue. Sometimes you might be able to manually find appropriate position even if master.info file is out of sync but I would not be basing my failure handling scenarios.

Cold Start - If you restart MySQL server its caches (key_buffer, innodb_buffer_pool, query_cache,table_cache) are cleaned, so may be OS caches. This may reduce performance dramatically. So if you're bringing server back after crash you might want to populate caches. For MyISAM key_cache this can be done by using LOAD INDEX INTO CACHE statement, for other storage engines it can be done by issuing large index scan queries. Full table scan queries allow to preload table data ether in storage engine caches or in OS cache. You can save these into .sql file and use –init-file to make sure it is run on startup. The other approach is to prime server with real servers (ie clone queries from other slave) before putting traffic to it.
In case application is not highly available so there is only one server you might with to start serving only some users initially (returning error to others) and gradually increase the load as server warms up. This may sound strange but makes a lot of sense as not only waiting for pages which never load is more frustrating for users than getting honest "try again later" message, but also - warmup takes longer time on extreme load.

Innodb statistics - Unlike MyISAM Innodb does not store index cardinality in tables, instead it computes them on first table access after startup. This may take significant time if you have very large number of tables (Some users have hundreds of thousands of tables per database host). This one is pretty much part of cold start problems but I wanted to point out it separately. To warmup this data you might run select 1 from _table_ limit 1 for each table or any other statement - it is table open which is important.

There are other problems which you may experience related to MySQL Crash Recovery - Restoring data from backup, corrupted Innodb tablespace recovery etc but I should write about them some other time.

Procédures stockées

Création

```
CREATE PROCEDURE proc1(IN parametre VARCHAR(10))
BEGIN
        SELECT COUNT(*) FROM table WHERE def=parametre;
END;
```

Paramètres

Les paramètres peuvent être de forme IN (paramètre entrant, comme pour une fonction), OUT (paramètre sortant, récupérable par le reste du code SQL) et INOUT (les deux à la fois), et un type de donnée (ici, il s'agit d'un entier, integer).

Utilisation

```
CALL proc1("test");
```

Déclaration de variables

Les procédures stockées peuvent déclarer leurs propres variables, et disposent également de la syntaxe SELECT..INTO, qui permet de stocker des colonnes choisies directement au sein de variables internes ou de session.

```
CREATE PROCEDURE sp1 (x VARCHAR(5))
BEGIN
DECLARE xname VARCHAR(5) DEFAULT 'bob';
DECLARE newname VARCHAR(5);
DECLARE xid INT;
SELECT xname,id INTO newname,xid
FROM table1 WHERE xname = xname;
SELECT newname;
END;
```

Incompatibilités suite à la migration vers MySql 5

C API change: mysql_shutdown() now requires a second argument. This is a source-level incompatibility that affects how you compile client programs; it does not affect the ability of compiled clients to communicate with older servers. (5.0.1)

SHOW STATUS now shows the session (thread-specific) status variables and SHOW GLOBAL STATUS shows the status variables for the whole server. (5.0.2)

A consequence of the change in handling of the DECIMAL and NUMERIC fixed-point data types is that the server is more strict to follow standard SQL. For example, a data type of DECIMAL(3,1) stores a maximum value of 99.9. Previously, the server allowed larger numbers to be stored. That is, it stored a value such as 100.0 as 100.0. Now the server clips 100.0 to the maximum allowable value of 99.9. If you have tables that were created before MySQL 5.0.3 and that contain floating-point data not strictly legal for the data type, you should alter the data types of those columns. (5.0.3 & 4.1.2)

For user-defined functions, exact-value decimal arguments such as 1.3 or DECIMAL column values were passed as REAL_RESULT values prior to MySQL 5.0.3. As of 5.0.3, they are passed as strings with a type of DECIMAL_RESULT. (5.0.3)

The C API ER_WARN_DATA_TRUNCATED warning symbol was renamed to WARN_DATA_TRUNCATED. (5.0.3)

MyISAM and InnoDB tables created with DECIMAL columns in MySQL 5.0.3 to 5.0.5 will appear corrupt after an upgrade to MySQL 5.0.6. Dump such tables with mysqldump before upgrading, and then reload them after upgrading. (The same incompatibility will occur for these tables created in MySQL 5.0.6 after a downgrade to MySQL 5.0.3 to 5.0.5.) (5.0.6)

The behavior of LOAD DATA INFILE and SELECT ... INTO OUTFILE has changed when the FIELDS TERMINATED BY and FIELDS ENCLOSED BY values both are empty. Formerly, a column was read or written the display width of the column. For example, INT(4) was read or written using a field with a width of 4. Now columns are read and written using a field width wide enough to hold all values in the field. However, data files written before this change was made might not be reloaded correctly with LOAD DATA INFILE for MySQL 4.1.12 and up. This change also affects data files read by mysqlimport and written by mysqldump --tab, which use LOAD DATA INFILE and SELECT ... INTO OUTFILE. (5.0.6 & 4.1.12)

Previously, conversion of DATETIME values to numeric form by adding zero produced a result in YYYYMMDDHHMMSS format. The result of DATETIME+0 is now in YYYYMMDDHHMMSS.000000 format. (5.0.8 & 4.1.13)

The namespace for triggers has changed. Previously, trigger names had to be unique per table. Now they must be unique within the schema (database). An implication of this change is that DROP TRIGGER syntax now uses a schema name instead of a table name (schema name is optional and, if omitted, the current schema will be used). (5.0.10)

Beginning with MySQL 5.0.12, natural joins and joins with USING, including outer join variants, are processed according to the SQL:2003 standard. The changes include elimination of redundant output columns for NATURAL joins and joins specified with a USING clause and proper ordering of output columns. The precedence of the comma operator also now is lower compared to JOIN. (5.0.12)

A lock wait timeout caused InnoDB to roll back the entire current transaction. Now it rolls back only the most recent SQL statement. (5.0.13)

For BINARY columns, the pad value and how it is handled has changed. The pad value for inserts now is 0x00 rather than space, and there is no stripping of the pad value for selects. (5.0.15)

The CHAR() function now returns a binary string rather than a string in the connection character set. An optional USING charset clause may be used to produce a result in a specific character set instead. Also, arguments larger than 256 produce multiple characters. They are no longer interpreted modulo 256 to produce a single character each. (5.0.15)

The InnoDB storage engine no longer ignores trailing spaces when comparing BINARY or VARBINARY column values. This means that (for example) the binary values 'a' and 'a ' are now regarded as unequal any time they are compared, as they are in MyISAM tables. (5.0.19)

Effacer les doublons

```
select day, MIN(id)
from test
group by day
having count(*) > 1

select bad_rows.*
from test as bad_rows
   inner join (
      select day, MIN(id) as min_id
      from test
      group by day
      having count(*) > 1
   ) as good_rows on good_rows.day = bad_rows.day
      and good_rows.min_id <> bad_rows.id;

delete bad_rows.*
from test as bad_rows
   inner join (
      select day, MIN(id) as min_id
      from test
      group by day
      having count(*) > 1
   ) as good_rows on good_rows.day = bad_rows.day
      and good_rows.min_id <> bad_rows.id;

delete test_outer.*
from test as test_outer
where exists(
   select *
   from test as test_inner
   where test_inner.day = test_outer.day
   group by day
   having count(*) > 1
      and min(test_inner.id) <> test_outer.id
);

CREATE TABLE new_fruits ...;

INSERT INTO new_fruits(fruit)
   SELECT DISTINCT fruit FROM fruits;

DROP TABLE fruits;

RENAME TABLE new_fruits fruits;

set @num := 0, @type := '';

delete from fruits
where greatest(0,
   @num := if(type = @type, @num + 1, 0),
   least(0, length(@type := type))) > 1
order by type;

set @num := 0;

select @type := type, @num := count(*)
   from fruits
   group by type
   having count(*) > 1
   limit 1;

while @num > 0

   delete from fruits where type = 'type'
      limit @num - 1;

   set @num := 0;
   select @type := type, @num := count(*)
      from fruits
      group by type
      having count(*) > 1
      limit 1;

end while
```

Sauvegarde

Script de sauvegarde sans interruption de service sur une machine esclave

```
#!/bin/sh

datum=`/bin/date +%Y%m%d-%H`

/usr/bin/mysqladmin --user=root --password=yourrootsqlpassword stop-slave

/usr/bin/mysqldump --user=root --password=yourrootsqlpassword --lock-all-tables \
      --all-databases > /home/sqlbackup/backup-${datum}.sql

/usr/bin/mysqladmin --user=root --password=yourrootsqlpassword start-slave

for file in "$( /usr/bin/find /home/sqlbackup -type f -mtime +2 )"
do
  /bin/rm -f $file
done

exit 0
```

Outils de sauvegarde

MySqlDump

mysqldump ––routines ––triggers ––single-transaction ––master-data=2 -p databasename > databasename.dmp
mysql -p < databasename.dmp

Lvm

mylvmbackup

Zmanda Recovery Manager

mysqlhotcopy (MyISam ou Isam) sous Unix

Utilitaire de sauvegarde à chaud

mysqlsnapshot

InnoDb Hot Backup

Annexe : variables MySql

Table 1: MySQL Variables

Variable Name	Variable Type	SET OFFLINE ONLY	SET GLOBAL	SET SESSION
auto_increment_increment	numeric	OFFLINE		
auto_increment_offset	numeric	OFFLINE		
autocommit	boolean			SESSION
automatic_sp_privileges	boolean		GLOBAL	
back_log	numeric	OFFLINE		
basedir	string	OFFLINE		
bdb_cache_size	numeric	OFFLINE		
bdb_home	string	OFFLINE		
bdb_log_buffer_size	numeric	OFFLINE		
bdb_logdir	string	OFFLINE		
bdb_max_lock	numeric	OFFLINE		
bdb_shared_data	boolean	OFFLINE		
bdb_tmpdir	string	OFFLINE		
big_tables	boolean			SESSION
binlog_cache_size	numeric		GLOBAL	
bulk_insert_buffer_size	numeric		GLOBAL	SESSION
character_set_client	string		GLOBAL	SESSION
character_set_connection	string		GLOBAL	SESSION
character_set_database	string	OFFLINE		
character_set_filesystem	string		GLOBAL	SESSION
character_set_results	string		GLOBAL	SESSION

character_set_server	string		GLOBAL	SESSION
character_set_system	string	OFFLINE		
character_sets_dir	string	OFFLINE		
collation_connection	string		GLOBAL	SESSION
collation_database	string	OFFLINE		
collation_server	string		GLOBAL	SESSION
completion_type	numeric		GLOBAL	SESSION
concurrent_insert	numeric		GLOBAL	
connect_timeout	numeric		GLOBAL	
datadir	string	OFFLINE		
date_format	–	OFFLINE		
datetime_format	–	OFFLINE		
default_week_format	numeric		GLOBAL	SESSION
delay_key_write	OFF | ON | ALL		GLOBAL	
delayed_insert_limit	numeric		GLOBAL	
delayed_insert_timeout	numeric		GLOBAL	
delayed_queue_size	numeric		GLOBAL	
div_precision_increment	numeric		GLOBAL	SESSION
engine_condition_pushdown	boolean		GLOBAL	SESSION
error_count	numeric			SESSION
expire_logs_days	numeric		GLOBAL	
flush	boolean		GLOBAL	
flush_time	numeric		GLOBAL	
foreign_key_checks	boolean			SESSION
ft_boolean_syntax	string		GLOBAL	
ft_max_word_len	numeric	OFFLINE		
ft_min_word_len	numeric	OFFLINE		
ft_query_expansion_limit	numeric	OFFLINE		
ft_stopword_file	string	OFFLINE		
group_concat_max_len	numeric		GLOBAL	SESSION
have_archive	boolean	OFFLINE		
have_bdb	boolean	OFFLINE		
have_blackhole_engine	boolean	OFFLINE		
have_compress	boolean	OFFLINE		
have_crypt	boolean	OFFLINE		
have_csv	boolean	OFFLINE		
have_dynamic_loading	boolean	OFFLINE		
have_example_engine	boolean	OFFLINE		
have_federated_engine	boolean	OFFLINE		
have_geometry	boolean	OFFLINE		
have_innodb	boolean	OFFLINE		
have_isam	boolean	OFFLINE		
have_merge_engine	boolean	OFFLINE		
have_ndbcluster	boolean	OFFLINE		
have_openssl	boolean	OFFLINE		
have_query_cache	boolean	OFFLINE		
have_raid	boolean	OFFLINE		
have_rtree_keys	boolean	OFFLINE		
have_symlink	boolean	OFFLINE		
init_connect	string	OFFLINE		
init_file	string	OFFLINE		
init_slave	string	OFFLINE		
innodb_additional_mem_pool_size	numeric	OFFLINE		
innodb_autoextend_increment	numeric		GLOBAL	
innodb_buffer_pool_awe_mem_mb	numeric	OFFLINE		
innodb_buffer_pool_size	numeric	OFFLINE		
innodb_checksums	boolean	OFFLINE		
innodb_commit_concurrency	numeric		GLOBAL	

innodb_concurrency_tickets	numeric		GLOBAL	
innodb_data_file_path	string	OFFLINE		
innodb_data_home_dir	string	OFFLINE		
innodb_doublewrite	boolean	OFFLINE		
innodb_fast_shutdown	0,1 or 2	OFFLINE		
innodb_file_io_threads	numeric	OFFLINE		
innodb_file_per_table	boolean	OFFLINE		
innodb_flush_log_at_trx_commit	0,1 or 2	OFFLINE		
innodb_flush_method	Enum	OFFLINE		
innodb_force_recovery	1 to 6	OFFLINE		
innodb_lock_wait_timeout	numeric	OFFLINE		
innodb_locks_unsafe_for_binlog	boolean	OFFLINE		
innodb_log_arch_dir	string	OFFLINE		
innodb_log_archive	–	OFFLINE		
innodb_log_buffer_size	numeric	OFFLINE		
innodb_log_file_size	numeric	OFFLINE		
innodb_log_files_in_group	numeric	OFFLINE		
innodb_log_group_home_dir	string	OFFLINE		
innodb_max_dirty_pages_pct	numeric		GLOBAL	
innodb_max_purge_lag	numeric		GLOBAL	
innodb_mirrored_log_groups	numeric	OFFLINE		
innodb_open_files	numeric	OFFLINE		
innodb_support_xa	boolean		GLOBAL	SESSION
innodb_sync_spin_loops	numeric		GLOBAL	
innodb_table_locks	boolean		GLOBAL	SESSION
innodb_thread_concurrency	numeric		GLOBAL	
innodb_thread_sleep_delay	numeric		GLOBAL	
interactive_timeout	numeric		GLOBAL	SESSION
join_buffer_size	numeric		GLOBAL	SESSION
key_buffer_size	numeric		GLOBAL	
key_cache_age_threshold	numeric	OFFLINE		
key_cache_block_size	numeric	OFFLINE		
key_cache_division_limit	1 to 100	OFFLINE		
language	string	OFFLINE		
large_files_support	boolean	OFFLINE		
large_page_size	numeric	OFFLINE		
large_pages	boolean	OFFLINE		
lc_time_names	string		GLOBAL	SESSION
license	string	OFFLINE		
local_infile	boolean		GLOBAL	
locked_in_memory	boolean	OFFLINE		
log	boolean	OFFLINE		
log_bin	boolean	OFFLINE		
log_bin_trust_function_creators	boolean		GLOBAL	
log_error	string	OFFLINE		
log_queries_not_using_indexes	boolean		GLOBAL	
log_slave_updates	boolean	OFFLINE		
log_slow_queries	boolean	OFFLINE		
log_warnings	numeric		GLOBAL	
long_query_time	numeric		GLOBAL	SESSION
low_priority_updates	boolean		GLOBAL	SESSION
lower_case_file_system	boolean	OFFLINE		
lower_case_table_names	numeric	OFFLINE		
max_allowed_packet	numeric		GLOBAL	SESSION
max_binlog_cache_size	numeric		GLOBAL	
max_binlog_size	numeric		GLOBAL	
max_connect_errors	numeric		GLOBAL	
max_connections	numeric		GLOBAL	

max_delayed_threads	numeric		GLOBAL	
max_error_count	numeric		GLOBAL	SESSION
max_heap_table_size	numeric		GLOBAL	SESSION
max_insert_delayed_threads	numeric		GLOBAL	
max_join_size	numeric		GLOBAL	SESSION
max_length_for_sort_data	numeric	OFFLINE		
max_prepared_stmt_count	numeric		GLOBAL	
max_relay_log_size	numeric		GLOBAL	
max_seeks_for_key	numeric		GLOBAL	SESSION
max_sort_length	numeric		GLOBAL	SESSION
max_sp_recursion_depth	numeric	OFFLINE		
max_tmp_tables	numeric		GLOBAL	SESSION
max_user_connections	numeric		GLOBAL	
max_write_lock_count	numeric		GLOBAL	
multi_range_count	numeric		GLOBAL	SESSION
myisam_data_pointer_size	numeric		GLOBAL	
myisam_max_sort_file_size	numeric		GLOBAL	SESSION
myisam_recover_options	boolean	OFFLINE		
myisam_repair_threads	numeric		GLOBAL	SESSION
myisam_sort_buffer_size	numeric		GLOBAL	SESSION
myisam_stats_method	enum		GLOBAL	SESSION
net_buffer_length	numeric		GLOBAL	SESSION
net_read_timeout	numeric		GLOBAL	SESSION
net_retry_count	numeric		GLOBAL	SESSION
net_write_timeout	numeric		GLOBAL	SESSION
new	boolean	OFFLINE		
old_passwords	boolean		GLOBAL	SESSION
open_files_limit	numeric	OFFLINE		
optimizer_prune_level	numeric		GLOBAL	SESSION
optimizer_search_depth	numeric		GLOBAL	SESSION
pid_file	string	OFFLINE		
port	numeric	OFFLINE		
preload_buffer_size	numeric		GLOBAL	SESSION
prepared_stmt_count	numeric	OFFLINE		
protocol_version	numeric	OFFLINE		
query_alloc_block_size	numeric		GLOBAL	SESSION
query_cache_limit	numeric		GLOBAL	
query_cache_min_res_unit	numeric	OFFLINE		
query_cache_size	numeric		GLOBAL	
query_cache_type	enumeration		GLOBAL	SESSION
query_cache_wlock_invalidate	boolean		GLOBAL	SESSION
query_prealloc_size	numeric		GLOBAL	SESSION
range_alloc_block_size	numeric		GLOBAL	SESSION
read_buffer_size	numeric		GLOBAL	SESSION
read_only	numeric		GLOBAL	
read_rnd_buffer_size	numeric		GLOBAL	SESSION
relay_log_purge	boolean	OFFLINE		
relay_log_space_limit	numeric	OFFLINE		
rpl_recovery_rank	numeric		GLOBAL	
secure_auth	boolean		GLOBAL	
server_id	numeric		GLOBAL	
skip_external_locking	boolean	OFFLINE		
skip_networking	boolean	OFFLINE		
skip_show_database	boolean	OFFLINE		
slave_compressed_protocol	boolean		GLOBAL	
slave_load_tmpdir	string	OFFLINE		
slave_net_timeout	numeric		GLOBAL	
slave_skip_errors	boolean	OFFLINE		

slave_transaction_retries	numeric		GLOBAL	
slow_launch_time	numeric		GLOBAL	
socket	string	OFFLINE		
sort_buffer_size	numeric		GLOBAL	SESSION
sql_auto_is_null	boolean			SESSION
sql_big_selects	boolean			SESSION
sql_big_tables	boolean			SESSION
sql_buffer_result	boolean			SESSION
sql_log_bin	boolean			SESSION
sql_log_off	boolean			SESSION
sql_log_update	boolean			SESSION
sql_low_priority_updates	boolean		GLOBAL	SESSION
sql_max_join_size	numeric		GLOBAL	SESSION
sql_mode	enumeration		GLOBAL	SESSION
sql_notes	boolean			SESSION
sql_quote_show_create	boolean			SESSION
sql_safe_updates	boolean			SESSION
sql_select_limit	numeric			SESSION
sql_slave_skip_counter	numeric		GLOBAL	
sql_warnings	boolean			SESSION
ssl_ca	string	OFFLINE		
ssl_capath	string	OFFLINE		
ssl_cert	string	OFFLINE		
ssl_cipher	string	OFFLINE		
ssl_key	string	OFFLINE		
storage_engine	enumeration		GLOBAL	SESSION
sync_binlog	numeric		GLOBAL	
sync_frm	boolean		GLOBAL	
system_time_zone	string	OFFLINE		
table_cache	numeric		GLOBAL	
table_lock_wait_timeout	numeric	OFFLINE		
table_type	enumeration		GLOBAL	SESSION
thread_cache_size	numeric		GLOBAL	
thread_stack	numeric	OFFLINE		
time_format	—	OFFLINE		
time_zone	string		GLOBAL	SESSION
timed_mutexes	boolean	OFFLINE		
tmp_table_size	enumeration		GLOBAL	SESSION
tmpdir	string	OFFLINE		
transaction_alloc_block_size	numeric		GLOBAL	SESSION
transaction_prealloc_size	numeric		GLOBAL	SESSION
tx_isolation	enumeration		GLOBAL	SESSION
updatable_views_with_limit	enumeration		GLOBAL	SESSION
version	string	OFFLINE		
version_bdb	string	OFFLINE		
version_comment	string	OFFLINE		
version_compile_machine	string	OFFLINE		
version_compile_os	numeric	OFFLINE		
wait_timeout	numeric		GLOBAL	SESSION

Table 2: The comment column

Variable Name	COMMENT
auto_increment_increment	Controls the interval between successive column values
auto_increment_offset	Determines the starting point for the AUTO_INCREMENT column value
autocommit	Statement forms a single transaction on its own
automatic_sp_privileges	Server automatically grants the EXECUTE and ALTER ROUTINE privileges to the creator of a stored routine, if the user cannot already execute and alter or drop the routine

back_log	Number of outstanding connection requests MySQL can have
basedir	MySQL installation base directory
bdb_cache_size	Size of the buffer allocated for caching indexes and rows for BDB tables
bdb_home	Base directory for BDB tables
bdb_log_buffer_size	Size of the buffer allocated for caching indexes and rows for BDB tables
bdb_logdir	Directory where the BDB storage engine writes its log files
bdb_max_lock	Maximum number of locks that can be active for a BDB table
bdb_shared_data	Start Berkeley DB in multi-process mode
bdb_tmpdir	BDB temporary file directory
big_tables	Exactly the same as using SQL_BIG_TABLES for all queries
binlog_cache_size	Size of the cache to hold the SQL statements for the binary log during a transaction
bulk_insert_buffer_size	Limits the size of the cache tree in bytes per thread
character_set_client	Character set for statements that arrive from the client
character_set_connection	Character set used for literals that do not have a character set introducer and for number-to-string conversion
character_set_database	Character set used by the default database
character_set_filesystem	Used to interpret string literals that refer to filenames
character_set_results	Character set used for returning query results to the client
character_set_server	Server's default character set
character_set_system	Character set used by the server for storing identifiers
character_sets_dir	Directory where character sets are installed
collation_connection	Collation of the connection character set
collation_database	Collation used by the default database
collation_server	Server's default collation
completion_type	Transaction completion type
concurrent_insert	Allows INSERT and SELECT statements to run concurrently for MyISAM tables that have no free blocks in the middle of the data file
connect_timeout	Number of seconds that the mysqld server waits for a connect packet before responding with Bad handshake
datadir	MySQL data directory
date_format	Not implemented
datetime_format	Not implemented
default_week_format	Default mode value to use for the WEEK() function
delay_key_write	ONLY MyISAM. If DELAY_KEY_WRITE is enabled for a table, the key buffer is not flushed for the table on every index update, but only when the table is closed.
delayed_insert_limit	After inserting <delayed_insert_limit> delayed rows, the INSERT DELAYED handler thread checks whether there are any SELECT statements pending. If so, it allows them to execute before continuing to insert delayed rows.
delayed_insert_timeout	How many seconds an INSERT DELAYED handler thread should wait for INSERT statements before terminating
delayed_queue_size	Per-table limit on the number of rows to queue when handling INSERT DELAYED statements
div_precision_increment	Number of digits of precision by which to increase the result of division operations performed with the / operator
engine_condition_pushdown	Applies to NDB
error_count	Read-only variable
expire_logs_days	Number of days for automatic binary log removal
flush	If ON, the server flushes (synchronizes) all changes to disk after each SQL statement.
flush_time	All tables are closed every flush_time seconds to free up resources and synchronize unflushed data to disk
foreign_key_checks	Do not check for foreign key problems
ft_boolean_syntax	List of operators supported by boolean full-text searches performed using IN BOOLEAN MODE
ft_max_word_len	Maximum length of the word to be included in a FULLTEXT index
ft_min_word_len	Minimum length of the word to be included in a FULLTEXT index
ft_query_expansion_limit	Number of top matches to use for full-text searches performed using WITH QUERY EXPANSION
ft_stopword_file	File from which to read the list of stopwords for full-text searches
group_concat_max_len	Maximum allowed result length for the GROUP_CONCAT() function
have_archive	YES if mysqld supports ARCHIVE tables, NO if not.
have_bdb	YES if mysqld supports BDB tables. DISABLED if –skip-bdb is used
have_blackhole_engine	YES if mysqld supports BLACKHOLE tables, NO if not
have_compress	YES if the zlib compression library is available to the server, NO if not. If not, the COMPRESS() and UNCOMPRESS() functions cannot be used

have_crypt	YES if the crypt() system call is available to the server, NO if not. If not, the ENCRYPT() function cannot be used
have_csv	YES if mysqld supports CSV tables, NO if not
have_dynamic_loading	YES if mysqld can dynamically load data
have_example_engine	YES if mysqld supports EXAMPLE tables, NO if not
have_federated_engine	YES if mysqld supports FEDERATED tables, NO if not
have_geometry	YES if the server supports spatial data types, NO if not
have_innodb	YES if mysqld supports InnoDB tables. DISABLED if –skip-innodb is used
have_isam	For backward compatibility. Always NO because ISAM tables are no longer supported.
have_merge_engine	YES if mysqld supports MERGE tables. DISABLED if –skip-merge is used
have_ndbcluster	YES if mysqld supports NDB Cluster tables. DISABLED if –skip-ndbcluster is used
have_openssl	YES if mysqld supports SSL connections, NO if not
have_query_cache	YES if mysqld supports the query cache, NO if not
have_raid	For backward compatibility. Always NO because RAID tables are no longer supported
have_rtree_keys	YES if RTREE indexes are available, NO if not
have_symlink	YES if symbolic link support is enabled, NO if not
init_connect	String to be executed by the server for each client that connects
init_file	Name of the file specified with the –init-file option when you start the server. This should be a file containing SQL statements that you want the server to execute when it starts
init_slave	String to be executed by a slave server each time the SQL thread starts
innodb_additional_mem_pool_size	Size in bytes of a memory pool InnoDB uses to store data dictionary information and other internal data structures
innodb_autoextend_increment	Increment size (in MB) for extending the size of an auto-extending tablespace when it becomes full
innodb_buffer_pool_awe_mem_mb	Size of the buffer pool (in MB), if it is placed in the AWE memory. Relevant only in 32-bit Windows
innodb_buffer_pool_size	Size in bytes of the memory buffer InnoDB uses to cache data and indexes of its tables
innodb_checksums	InnoDB can use checksum validation on all pages read from the disk to ensure extra fault tolerance against broken hardware or data files
innodb_commit_concurrency	Number of threads that can commit at the same time. A value of 0 disables concurrency control.
innodb_concurrency_tickets	Number of threads that can enter InnoDB concurrently is determined by the innodb_thread_concurrency variable
innodb_data_file_path	Paths to individual data files and their sizes
innodb_data_home_dir	Common part of the directory path for all InnoDB data files
innodb_doublewrite	InnoDB stores all data twice, first to the doublewrite buffer, and then to the actual data files
innodb_fast_shutdown	Full purge and an insert buffer merge before a shutdown OR NOT
innodb_file_io_threads	Number of file I/O threads in InnoDB
innodb_file_per_table	If this variable is enabled, InnoDB creates each new table using its own .ibd file for storing data and indexes, rather than in the shared tablespace
innodb_flush_log_at_trx_commit	See documentation
innodb_flush_method	See documentation
innodb_force_recovery	Crash recovery mode
innodb_lock_wait_timeout	Timeout in seconds an InnoDB transaction may wait for a lock before being rolled back
innodb_locks_unsafe_for_binlog	Controls next-key locking in InnoDB searches and index scans
innodb_log_arch_dir	Path to arch dir
innodb_log_archive	Present for historical reasons, but unused
innodb_log_buffer_size	Size in bytes of the buffer that InnoDB uses to write to the log files on disk
innodb_log_file_size	Size in bytes of each log file in a log group
innodb_log_files_in_group	Number of log files in the log group
innodb_log_group_home_dir	Directory path to the InnoDB log files
innodb_max_dirty_pages_pct	Main thread in InnoDB tries to write pages from the buffer pool so that the percentage of dirty (not yet written) pages will not exceed this value
innodb_max_purge_lag	Controls how to delay INSERT, UPDATE and DELETE operations when the purge operations are lagging
innodb_mirrored_log_groups	Number of identical copies of log groups to keep for the database
innodb_open_files	Relevant only if you use multiple tablespaces in InnoDB. It specifies the maximum number of .ibd files that InnoDB can keep open at one time
innodb_support_xa	Enables InnoDB support for two-phase commit in XA transactions
innodb_sync_spin_loops	Number of times a thread waits for an InnoDB mutex to be freed before the thread is suspended
innodb_table_locks	If 1, means that LOCK TABLES causes InnoDB to lock a table internally if AUTOCOMMIT=0
innodb_thread_concurrency	InnoDB tries to keep the number of operating system threads concurrently inside InnoDB less than or equal to this limit
innodb_thread_sleep_delay	How long InnoDB threads sleep before joining the InnoDB queue, in microseconds

interactive_timeout	Number of seconds the server waits for activity on an interactive connection before closing it
join_buffer_size	Size of the buffer used for joins that do not use indexes and thus perform full table scans
key_buffer_size	Size of the buffer used for index blocks (also known as the key cache)
key_cache_age_threshold	Controls the demotion of buffers from the hot sub-chain of a key cache to the warm sub-chain
key_cache_block_size	Size in bytes of blocks in the key cache
key_cache_division_limit	Division point between the hot and warm sub-chains of the key cache buffer chain. The value is the percentage of the buffer chain to use for the warm sub-chain
language	Language used for error messages
large_files_support	Whether mysqld was compiled with options for large file support
large_page_size	
large_pages	Whether large page support is enabled
lc_time_names	Specifies the locale that controls the language used to display day and month names and abbreviations
license	Type of license the server has
local_infile	Whether LOCAL is supported for LOAD DATA INFILE statements
locked_in_memory	Whether mysqld was locked in memory with –memlock
log	Whether logging of all statements to the general query log is enabled
log_bin	Whether the binary log is enabled
log_bin_trust_function_creators	Applies when binary logging is enabled. Controls whether stored function creators can be trusted not to create stored functions that will cause unsafe events to be written to the binary log.
log_error	Location of the error log
log_queries_not_using_indexes	Whether queries that do not use indexes are logged to the slow query log
log_slave_updates	Whether updates received by a slave server from a master server should be logged to the slave's own binary log
log_slow_queries	Whether slow queries should be logged
log_warnings	Whether to produce additional warning messages
long_query_time	If a query takes longer than this many seconds, the query is logged to the slow query log file
low_priority_updates	If set to 1, all INSERT, UPDATE, DELETE, and LOCK TABLE WRITE statements wait until there is no pending SELECT or LOCK TABLE READ on the affected table
lower_case_file_system	Describes the case sensitivity of filenames on the filesystem where the data directory is located
lower_case_table_names	If set to 1, table names are stored in lowercase on disk and table name comparisons are not case sensitive. If set to 2 table names are stored as given but compared in lowercase
max_allowed_packet	Maximum size of one packet or any generated/intermediate string
max_binlog_cache_size	If a multiple-statement transaction requires more than this many bytes of memory, the server generates a multi-statement transaction requiring more than 'max_binlog_cache_size' bytes of storage error
max_binlog_size	If a write to the binary log causes the current log file size to exceed the value of this variable, the server rotates the binary logs (closes the current file and opens the next one)
max_connect_errors	If there are more than this number of interrupted connections from a host, that host is blocked from further connections
max_connections	Number of simultaneous client connections allowed
max_delayed_threads	Do not start more than this number of threads to handle INSERT DELAYED statements
max_error_count	Maximum number of error, warning, and note messages to be stored for display by the SHOW ERRORS and SHOW WARNINGS statements
max_heap_table_size	Sets the maximum size to which MEMORY tables are allowed to grow
max_insert_delayed_threads	Synonym for max_delayed_threads
max_join_size	Do not allow SELECT statements that probably need to examine more than max_join_size rows (for single-table statements) or row combinations (for multiple-table statements) or that are likely to do more than max_join_size disk seeks
max_length_for_sort_data	Cutoff on the size of index values that determines which filesort algorithm to use
max_prepared_stmt_count	Limits the total number of prepared statements in the server
max_relay_log_size	If a write by a replication slave to its relay log causes the current log file size to exceed the value of this variable, the slave rotates the relay logs (closes the current file and opens the next one)
max_seeks_for_key	Limit the assumed maximum number of seeks when looking up rows based on a key
max_sort_length	Number of bytes to use when sorting BLOB or TEXT values
max_sp_recursion_depth	Number of times that a stored procedure may call itself
max_tmp_tables	Maximum number of temporary tables a client can keep open at the same time
max_user_connections	Maximum number of simultaneous connections allowed to any given MySQL account
max_write_lock_count	After this many write locks, allow some pending read lock requests to be processed in between
multi_range_count	Maximum number of ranges to send to a table handler at once during range selects
myisam_data_pointer_size	Default pointer size in bytes, to be used by CREATE TABLE for MyISAM tables when no MAX_ROWS option is specified
myisam_max_sort_file_size	Maximum size of the temporary file that MySQL is allowed to use while re-creating a MyISAM index

	(during REPAIR TABLE, ALTER TABLE, or LOAD DATA INFILE)
myisam_recover_options	Value of the –myisam-recover option
myisam_repair_threads	If greater than 1, MyISAM table indexes are created in parallel (each index in its own thread) during the Repair by sorting process
myisam_sort_buffer_size	Size of the buffer that is allocated when sorting MyISAM indexes during a REPAIR TABLE or when creating indexes with CREATE INDEX or ALTER TABLE
myisam_stats_method	How the server treats NULL values when collecting statistics about the distribution of index values for MyISAM tables
net_buffer_length	Each client thread is associated with a connection buffer and result buffer. Both begin with a size given by net_buffer_length but are dynamically enlarged up to max_allowed_packet bytes as needed
net_read_timeout	Number of seconds to wait for more data from a connection before aborting the read
net_retry_count	If a read on a communication port is interrupted, retry this many times before giving up
net_write_timeout	Number of seconds to wait for a block to be written to a connection before aborting the write
new	Retained for backward compatibility
old_passwords	Whether the server should use pre-4.1-style passwords for MySQL user accounts
open_files_limit	Number of files that the operating system allows mysqld to open
optimizer_prune_level	Controls the heuristics applied during query optimization to prune less-promising partial plans from the optimizer search space
optimizer_search_depth	Maximum depth of search performed by the query optimizer
pid_file	Pathname of the process ID (PID) file
port	Number of the port on which the server listens for TCP/IP connections
preload_buffer_size	Size of the buffer that is allocated when preloading indexes
prepared_stmt_count	Current number of prepared statements
protocol_version	Version of the client/server protocol used by the MySQL server
query_alloc_block_size	Allocation size of memory blocks that are allocated for objects created during statement parsing and execution
query_cache_limit	Don’t cache results that are larger than this number of bytes
query_cache_min_res_unit	Minimum size (in bytes) for blocks allocated by the query cache
query_cache_size	Amount of memory allocated for caching query results
query_cache_type	Sets the query cache type
query_cache_wlock_invalidate	Setting this variable to 1 causes acquisition of a WRITE lock for a table to invalidate any queries in the query cache that refer to the table. This forces other clients that attempt to access the table to wait while the lock is in effect.
query_prealloc_size	Size of the persistent buffer used for statement parsing and execution
range_alloc_block_size	Size of blocks allocated when doing range optimization
read_buffer_size	Each thread that does a sequential scan allocates a buffer of this size (in bytes) for each table it scans
read_only	When this variable is set to ON, the server allows no updates except from users that have the SUPER privilege, or (on a slave server) from updates performed by slave threads
read_rnd_buffer_size	When reading rows in sorted order following a key-sorting operation, the rows are read through this buffer to avoid disk seeks
relay_log_purge	Disables or enables automatic purging of relay log files as soon as they are no longer needed
relay_log_space_limit	Places an upper limit on the total size in bytes of all relay logs on the slave
rpl_recovery_rank	Variable is unused
secure_auth	Blocks connections from all accounts that have passwords stored in the old (pre-4.1) format
server_id	Used for replication to enable master and slave servers to identify themselves uniquely
skip_external_locking	This is OFF if mysqld uses external locking, ON if external locking is disabled
skip_networking	This is ON if the server allows only local (non-TCP/IP) connections
skip_show_database	Prevents users from using the SHOW DATABASES statement if they do not have the SHOW DATABASES privilege
slave_compressed_protocol	Whether to use compression of the slave/master protocol if both the slave and the master support it
slave_load_tmpdir	Name of the directory where the slave creates temporary files for replicating LOAD DATA INFILE statements
slave_net_timeout	Number of seconds to wait for more data from a master/slave connection before aborting the read
slave_skip_errors	Replication errors that the slave should skip (ignore)
slave_transaction_retries	If a replication slave SQL thread fails to execute a transaction, it automatically retries <slave_transaction_retries> times before stopping with an error
slow_launch_time	If creating a thread takes longer than this many seconds, the server increments the Slow_launch_threads status variable.
socket	On Unix platforms, this variable is the name of the socket file that is used for local client connections
sort_buffer_size	Each thread that needs to do a sort allocates a buffer of this size. Increase this value for faster ORDER BY or GROUP BY operations.

sql_auto_is_null	If set to 1 (the default), you can find the last inserted row for a table that contains an AUTO_INCREMENT column.
sql_big_selects	If set to 0, MySQL aborts SELECT statements that are likely to take a very long time to execute (that is, statements for which the optimizer estimates that the number of examined rows exceeds the value of max_join_size).
sql_big_tables	Renamed BIG_TABLES
sql_buffer_result	If set to 1, SQL_BUFFER_RESULT forces results from SELECT statements to be put into temporary tables.
sql_log_bin	If set to 0, no logging is done to the binary log for the client. The client must have the SUPER privilege to set this option.
sql_log_off	If set to 1, no logging is done to the general query log for this client. The client must have the SUPER privilege to set this option.
sql_log_update	Deprecated, and is mapped to SQL_LOG_BIN
sql_low_priority_updates	Renamed low_priority_updates
sql_max_join_size	Renamed max_join_size
sql_mode	Current server SQL mode
sql_notes	If set to 1 (the default), warnings of Note level are recorded. If set to 0, Note warnings are suppressed
sql_quote_show_create	If set to 1 (the default), the server quotes identifiers for SHOW CREATE TABLE and SHOW CREATE DATABASE statements. If set to 0, quoting is disabled
sql_safe_updates	If set to 1, MySQL aborts UPDATE or DELETE statements that do not use a key in the WHERE clause or a LIMIT clause
sql_select_limit	Maximum number of rows to return from SELECT statements
sql_slave_skip_counter	Number of events from the master that a slave server should skip
sql_warnings	Controls whether single-row INSERT statements produce an information string if warnings occur
ssl_ca	Path to a file with a list of trusted SSL CAs
ssl_capath	Path to a directory that contains trusted SSL CA certificates in PEM format
ssl_cert	Name of the SSL certificate file to use for establishing a secure connection
ssl_cipher	List of allowable ciphers to use for SSL encryption
ssl_key	Name of the SSL key file to use for establishing a secure connection
storage_engine	Default storage engine (table type)
sync_binlog	If the value is positive, the MySQL server synchronizes its binary log to disk (fdatasync()) after every sync_binlog writes to this binary log.
sync_frm	If set to 1, when any non-temporary table is created its .frm file is synchronized to disk (using fdatasync())
system_time_zone	Server system time zone
table_cache	Number of open tables for all threads
table_lock_wait_timeout	Specifies a wait timeout for table-level locks, in seconds
table_type	Synonym for storage_engine
thread_cache_size	How many threads the server should cache for reuse
thread_stack	Stack size for each thread
time_format	Not implemented
time_zone	Current time zone. Used to initialize the time zone for each client that connects.
timed_mutexes	Controls whether InnoDB mutexes are timed
tmp_table_size	Maximum size of in-memory temporary tables
tmpdir	Directory used for temporary files and temporary tables
transaction_alloc_block_size	Amount in bytes by which to increase a per-transaction memory pool that needs memory
transaction_prealloc_size	See documentation
tx_isolation	Default transaction isolation level
updatable_views_with_limit	Controls whether updates to a view can be made when the view does not contain all columns of the primary key defined in the underlying table, if the update statement contains a LIMIT clause
version	Version number for the server
version_bdb	BDB storage engine version
version_comment	Allows a comment to be specified when building MySQL
version_compile_machine	Type of machine or architecture on which MySQL was built
version_compile_os	Type of operating system on which MySQL was built
wait_timeout	Number of seconds the server waits for activity on a non-interactive connection before closing it

Documentation

Sites

http://www.howtoforge.com/back_up_mysql_dbs_without_interruptions