

# Découverte de MySQL, PostgreSQL et Oracle

---

## Résumé

Ce document explique aux administrateurs de bases de données les différents aspects à cerner lors de l'administration d'une base de données inconnue. Ce document est aussi utile aux administrateurs qui ne connaissent qu'un des logiciels présentés, afin de connaître les commandes équivalentes.

Sont présentés ici MySQL 5.1, PostgreSQL 8.1 et Oracle 10g installés sur une Debian Etch.

## Introduction

On présentera ici trois systèmes différents de gestion de base de données :

- MySQL 5.1, base de données libre très utilisée par les hébergeurs web pour ses bonnes performances et dont la version 5 comble les principaux manques (trigger, procédures stockées...). MySQL a la particularité de fournir plusieurs moteurs de bases de données (InnoDB, MyIsam, Berkeley DB...), qui ne fournissent pas les mêmes fonctionnalités ni les mêmes performances, afin de s'adapter au besoin. Par exemple, le moteur ARCHIVE est optimisé pour stocker et fournir des informations, mais il n'est pas possible de mettre à jour une ligne dans une table.
- PostgreSQL 8.1 est la base de données qui offre le plus de fonctionnalités en logiciel libre.
- Oracle 10g

## Conseils préliminaires

Voici quelques conseils qui permettent de gagner du temps lors de la découverte et la maintenance d'une base de données :

1. Demander s'il y a une fiche technique du serveur sur un partage de fichiers ou sur un wiki.
2. Interroger les personnes qui travaillent sur ce serveur pour savoir s'il y a des recommandations en plus par rapport à la documentation au cas où elle ne serait pas à jour.
3. Créer ou mettre à jour la documentation du serveur.
4. Lire la documentation officielle de la base de données et regarder les commentaires :
  - [\(fr\) MySQL 5.0](#) <sup>[1]</sup> | [\(en\) MySQL 5.1](#) <sup>[2]</sup>
  - [\(fr\) PostgreSQL 8.1.5](#) <sup>[3]</sup>
  - [\(en\) Oracle](#) <sup>[4]</sup>

## Détermination du système d'exploitation

- Unixes : `cat /etc/issue`
  - Linux:
    - Debian: on trouve ce fichier `/etc/debian_version` qui contient la version de Debian.
    - Redhat: `/etc/redhat_release`

Pour la suite du document, on considère que la base de données fonctionne sur Debian, mais la quasi totalité des explications dans la suite fonctionne avec les autres distributions Linux. Il sera indiqué quand la commande est spécifique à Debian.

## Détermination de l'environnement

Pour connaître le(s) processeur(s) :

```
# cat /proc/cpuinfo
```

Les partitions swap utilisées :

```
# cat /proc/swaps
```

La quantité de RAM disponible/utilisée :

```
# cat /proc/meminfo
```

La configuration réseau :

```
# ifconfig
```

## Services en cours de fonctionnement

Toutes les commandes sont à exécuter en root.

### Pour connaître la liste des processus qui fonctionnent :

```
# ps aux
```

### Pour connaître la liste des démons au démarrage de l'ordinateur :

Regarder le nom du logiciel de démarrage dans la liste des processus, c'est celui dont le PID vaut 1.

- Si vous obtenez `init` :
  - Regarder le chiffre entre `[]` derrière `init`, puis taper : `# ls /etc/rc[chiffre].d/`

Par exemple si le chiffre est 2 :

```
# ls /etc/rc2.d/
```

de manière générique :

```
# ls -la /etc/rc`runlevel` |awk '{print $2}'`.d
```

- Si vous obtenez `initng`, il faut taper :

```
# ng-update show
```

pour obtenir la liste des démons démarrés.

## Les logiciels installés | La façon dont ils sont installés

Il est possible d'installer de deux manières différentes un logiciel dans une distribution linux. Soit avec un paquetage soit avec les sources. L'intérêt du paquetage, c'est qu'il est beaucoup plus facile à installer et à mettre à jour que les sources. Par contre, plus de personnalisations du logiciel sont possibles grâce à une installation à partir des sources.

Dans debian, pour gérer les paquetages, on utilise le logiciel aptitude. Il permet d'installer, désinstaller et mettre à jour l'ensemble des logiciels présents sur le serveur. Dans le menu d'aptitude il est possible de rechercher un paquetage.

Si le paquetage de la base de données n'est pas installé, alors il est installé avec les sources. La plupart du temps, les logiciels installés à partir des sources se situent dans /usr/local/bin.

La façon dont est installé le logiciel détermine la procédure des mises à jour.

## Pour visualiser l'état de fonctionnement de la base de données

### MySQL

```
# /etc/init.d/mysqld status
```

Exemple de résultat pour une base en cours de fonctionnement :

```
/usr/bin/mysqladmin Ver 8.41 Distrib 5.0.18, for pc-linux-gnu on i486
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX
DataKonsult AB
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license

Server version          5.0.18-Debian_7-log
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/run/mysqld/mysqld.sock
Uptime:                 4 min 47 sec
```

```
Threads: 1 Questions: 157 Slow queries: 0 Opens: 0 Flush tables: 1
Open tables: 64 Queries per second avg: 0.547
```

**PostgreSQL**

```
# /etc/init.d/postgresql-8.1 status
```

Exemple de résultat pour une base en cours de fonctionnement :

```
Version Cluster  Port Status Owner  Data directory
          Log file
8.1      main      5432 online postgres /var/lib/postgresql/8.1/main
          /var/log/postgresql/postgresql-8.1-main.log
```

**Oracle**

```
# /etc/init.d/oracle status
```

On identifie la liste des instances Oracle en cours de fonctionnement avec la commande:

1. `ps -ef | grep pmon`

(Le processus PMON, Process Monitor, étant unique à chaque instance, et identifiant le SID de la base concernée)

**Comment démarrer/arrêter/redémarrer une base de données****MySQL**

- Avec init
  - Dans le cas d'une base de données classique :

```
# /etc/init.d/mysql (start|stop|restart|reload|force-reload)
```

- Dans le cas d'un noeud de cluster :

```
# /etc/init.d/mysql-ndb (start|stop|restart|reload|force-reload)
```

- Dans le cas d'un médiateur de cluster :

```
# /etc/init.d/mysql-ndb-mgm (start|stop|restart|reload|force-reload)
```

- Avec initng

```
# ngc --(start|stop|restart) mysql
```

**PostgreSQL**

- Avec init

Dans le cas d'une base de données :

```
# /etc/init.d/postgresql-8.1
(start|stop|restart|reload|force-reload|status|autovac-start|autovac-stop|autovac-restart)
```

- Avec initng

```
# ngc --(start|stop|restart) postgresql
```

## Oracle

- Avec init

Dans le cas d'une base de données :

```
# /etc/init.d/oracle (start|stop|restart)
```

- Avec initng

```
# ngc --(start|stop|restart) oracle
```

## Les fichiers de configuration

### MySQL

Tous les fichiers se trouvent dans le dossier `/etc/mysql` :

- *debian.cnf* : fichier de configuration généré automatiquement par Debian, contient le login de l'utilisateur système Debian dans MySQL ainsi que son mot de passe. Cet utilisateur permet à Debian de faire la maintenance des bases de MySQL.

#### Exemple :

```
Automatically generated for Debian scripts. DO NOT TOUCH!  
[client]  
host      = localhost  
user      = debian-sys-maint  
password  = XXXXXXXXXXXXXXXXXXXX  
socket    = /var/run/mysqld/mysqld.sock
```

- *debian-log-rotate.conf* : Définit la quantité de fichiers de logs à stocker.

#### Exemple :

```
KEEP_BINARY_LOGS=2
```

- *debian-start* : Script qui contient la liste des opérations de maintenance à effectuer lors du démarrage de MySQL.

#### Exemple :

```
#!/bin/bash  
#  
# This script is executed by "/etc/init.d/mysql" on every (re)start.  
#  
# Changes to this file will be preserved when updating the Debian  
package.  
#  
  
source /usr/share/mysql/debian-start.inc.sh  
  
MYADMIN="/usr/bin/mysqladmin --defaults-file=/etc/mysql/debian.cnf"  
MYCHECK="/usr/bin/mysqlcheck --defaults-file=/etc/mysql/debian.cnf"  
MYCHECK_SUBJECT="WARNING: mysqlcheck has found corrupt tables"  
MYCHECK_PARAMS="--all-databases --fast --silent"
```

```
# Put commands in the background if they should not block the server
start.
(
  mycheck;
) &

exit 0
```

- *my.cnf* : Tous les paramètres de configuration de MySQL comme les ports/sockets d'écoute, les interfaces réseaux autorisées, le tuning de la base, l'emplacement des logs et des fichiers de base de données... Enfin, ce fichier définit si le serveur MySQL est un noeud de cluster ou un le maître du cluster. Les commentaires dans ce fichier sont assez détaillés.

**Exemple :**

```
#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "/var/lib/mysql/my.cnf" to set server-specific options or
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with
ticks/quotes
# especially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket
location.
[client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

# This was formally known as [safe_mysqld]. Both versions are currently
parsed.
[mysqld_safe]
socket              = /var/run/mysqld/mysqld.sock
nice                = 0
```

```
[mysqld]
#
# * Basic Settings
#
user                = mysql
pid-file            = /var/run/mysqld/mysqld.pid
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
language            = /usr/share/mysql/english
skip-external-locking
#
# For compatibility to other Debian packages that still use
# libmysqlclient10 and libmysqlclient12.
old_passwords       = 1
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 127.0.0.1
#
# * Fine Tuning
#
key_buffer           = 16M
max_allowed_packet   = 16M
thread_stack         = 128K
#
# * Query Cache Configuration
#
query_cache_limit    = 1048576
query_cache_size     = 16777216
query_cache_type     = 1
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
#log                 = /var/log/mysql.log
#log                 = /var/log/mysql/mysql.log
#
# Error logging goes to syslog. This is a Debian improvement :)
#
# Here you can see queries with especially long duration
#log-slow-queries    = /var/log/mysql/mysql-slow.log
#
# The following can be used as easy to replay backup logs or for
```

```
replication.
#server-id          = 1
log-bin             = /var/log/mysql/mysql-bin.log
# See /etc/mysql/debian-log-rotate.conf for the number of files kept.
max_binlog_size    = 104857600
#binlog-do-db      = include_database_name
#binlog-ignore-db  = include_database_name
#
# * BerkeleyDB
#
# According to an MySQL employee the use of BerkeleyDB is now
discouraged
# and support for it will probably cease in the next versions.
skip-bdb
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# If you want to enable SSL support (recommended) read the manual or my
# HOWTO in /usr/share/doc/mysql-server/SSL-MINI-HOWTO.txt.gz
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem

[mysqldump]
quick
quote-names
max_allowed_packet = 16M

[mysql]
#no-auto-rehash      # faster start of mysql but no tab completion

[isamchk]
key_buffer           = 16M

#
# * NDB Cluster
#
# See /usr/share/doc/mysql-server-*/README.Debian for more information.
#
# The following configuration is read by the ndbd storage daemons,
# not from the ndb_mgmd management daemon.
```



```
#
# [MYSQL_CLUSTER]
# ndb-connectstring=127.0.0.1
```

### PostgreSQL

Les fichiers de configuration se trouvent dans `/usr/local/pgsql/data` ou `/etc/postgresql/8.1/main/`. PostgreSQL possède 3 fichiers de configuration:

- `pg_hba.conf`: fichier de configuration pour l'authentification des utilisateurs. Ce fichier est une suite d'enregistrements.

Les lignes vides sont ignorées. Chaque ligne est structurée d'une des manières suivantes:

```
local      database  user  auth-method  [auth-option]
host       database  user  CIDR-address  auth-method  [auth-option]
hostssl    database  user  CIDR-address  auth-method  [auth-option]
hostnossl  database  user  CIDR-address  auth-method  [auth-option]
host       database  user  IP-address    IP-mask      auth-method
[auth-option]
hostssl    database  user  IP-address    IP-mask      auth-method
[auth-option]
hostnossl  database  user  IP-address    IP-mask      auth-method
[auth-option]
```

- `pg_ident.conf`: fichier de configuration utilisé si l'on veut effectuer une authentification automatique avec le login utilisé lors du lancement de l'OS.

### Exemple :

```
# MAPNAME      IDENT-USERNAME  PG-USERNAME
omicron        bryanh          bryanh
omicron        ann             ann
# bob has user name robert on these machines
omicron        robert         bob
# bryanh can also connect as guest1
omicron        bryanh         guest1
```

- `postgresql.conf`: principal fichier de configuration. Toutes les informations propres au SGBD sont stockées dans ce fichier.

### Oracle

Par défaut, le fichier de configuration se trouve dans **`/home/oracle/admin/%ORACLE_SID%/pfile`**

et se nomme en général `init%ORACLE_SID%.ora`

*%ORACLE\_SID% est le nom de l'instance d'oracle utilisée*

Il contient en général tous les paramètres dont a besoin la base. **Exemple**

```
db_name=isima
db_files=1020
control_files=/home/oracle/admin/isima/isima.ora
db_file_multiblock_read_count = 8
```

```
db_block_buffers=200
shared_pool_size=6500000
log_checkpoint_interval=8000
processes=50
dml_locks=100
log_buffers=8192
sequence_cache_entries=10
sequence_cache_hash_buckets=10
#audit_trail=true
#timed_statistics=true
background_dump_dest=/home/oracle/rdbms80/trace
user_dump_dest=/home/oracle/rdbms80/trace
db_block_size=2048
compatible=8.0.4.0.0
...
```

## Localisation physique des données

1. Déterminer la localisation dans l'arborescence système des bases de données à partir du fichier de configuration. Par exemple pour MySQL, il se situe dans `/var/lib/mysql`
2. Taper :

```
# mount
```

pour connaître le point de montage des partitions.

### Exemple :

```
/dev/md0 on / type reiserfs (rw,notail)
/dev/sda6 on /home type ext3 (rw)
```

- Si le fichier du périphérique commence par `/dev/md`, alors il s'agit d'une partition en RAID software. Pour connaître la localisation physique des fichiers, il faut regarder le fichier `/etc/mdadm/mdadm.conf`

### Exemple :

```
DEVICE partitions
ARRAY /dev/md0 level=raid1 num-devices=2
UUID=8402604e:5b65bb45:f6c894f0:906c8b43
    devices=/dev/hda1,/dev/hdd1
MAILADDR example@example.com
```

Dans cet exemple, on peut voir que `/dev/md0` est un RAID réparti entre deux partitions, `hda1` et `hdd1`. L'adresse de l'administrateur à avertir en cas de défaillance d'une des deux partitions est également indiqué.

1. La commande :

```
# df -h
```

affiche l'espace disponible pour chaque partition

### Exemple :

Sys. de fich.	Tail.	Occ.	Disp.	%Occ.	Monté sur
/dev/md0	186G	163G	23G	88%	/
/dev/sda6	9,2G	1,7G	7,1G	20%	/home

Cette commande :

```
# du -hs /var/lib/mysql/*
```

affiche des informations précises sur la taille de chaque fichier.

## Interfaces de connexion à la base de données

Il est possible de gérer une base de données par différents moyens : Web, interface graphique, et ligne de commande. Il faut donc identifier les différentes méthodes possibles pour gérer la base de données.

### Web

#### MySQL

- PhpMyAdmin est l'interface web la plus utilisée pour gérer une base de données MySQL.

#### PostgreSQL

- PhpPgAdmin

#### Oracle

- isql\*plus
- sql\*Plus
- SQLPlus Worksheet

## Applications graphique

### Pour tous les serveurs de base de données

- Kexi - Outil KDE
- OpenOffice.org Base - Intégré à OpenOffice 2. permet de générer des schémas de la base de données.

## MySQL

- MySQL Administrator <sup>[5]</sup>

## PostgreSQL

- pgAdmin <sup>[6]</sup>

## Oracle

- Tora
- Oracle Enterprise Manager <sup>[7]</sup>

## Ligne de commande

### MySQL

```
# mysql
```

### PostgreSQL

```
# psql
```

### Oracle

```
# sqlplus
```

## Analyse interne de la base de données

On va maintenant déterminer les composantes internes de la base de données. Pour cela, on dispose d'une liste de commandes, très souvent en SQL. Voilà également quelques commandes utiles pour un administrateur de base de données.

## Généralités

### MySQL

Pour auditer un serveur MySQL, la commande SHOW permet d'afficher la plupart des informations utiles. (Documentation détaillée <sup>[8]</sup>)

## Utilisateurs - rôles - droits

### MySQL

(Documentation détaillée <sup>[9]</sup>)

- Liste des utilisateurs

```
USE mysql  
SELECT * FROM `user`
```

- Créer un utilisateur (Documentation détaillée <sup>[10]</sup>)

```
CREATE USER user [IDENTIFIED BY [PASSWORD] 'password']
```

- Supprimer un utilisateur (Documentation détaillée <sup>[11]</sup>)

```
DROP USER user;
```

## PostgreSQL

L'authentification dans PostgreSQL se fait via les rôles ou via les utilisateurs.

- Liste des rôles:

```
SELECT rolname FROM pg_roles;
```

- Créer un rôle: (Documentation détaillée <sup>[12]</sup>)

```
CREATE ROLE name [ [ WITH ] option [ ... ] ]
```

- Supprimer un rôle: (Documentation détaillée <sup>[13]</sup>)

```
DROP ROLE name;
```

- liste des utilisateurs

```
SELECT username FROM pg_user;
```

ou en général on utilise la métacommande \du

- Création d'un utilisateur (Documentation détaillée <sup>[14]</sup>)

```
CREATE USER nom [ [ WITH ]
    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | CREATEUSER | NOCREATEUSER
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | CONNECTION LIMIT limite_connexion
  | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'motdepasse'
  | VALID UNTIL 'dateheure'
  | IN ROLE nomrole [, ...]
  | IN GROUP nomrole [, ...]
  | ROLE nomrole [, ...]
  | ADMIN nomrole [, ...]
  | USER nomrole [, ...]
  | SYSID uid
```

- Effacer un utilisateur

```
DROP USER
```

- modifier un utilisateur

```
ALTER USER
```

## Oracle

### • Les Utilisateurs

- Afficher les utilisateurs

```
SELECT * FROM DBA_USERS
```

- Créer un utilisateur

```
CREATE USER nom  
IDENTIFIED BY mot_de_passe  
DEFAULT TABLESPACE tablespace_par_defaut  
TEMPORARY TABLESPACE tablespace_temporaire  
QUOTA quota ON tablespace
```

- Supprimer un utilisateur

```
DROP USER utilisateur [CASCADE]
```

- Modifier un utilisateur

```
ALTER USER utilisateur
```

### • Les rôles et les privilèges

- Afficher les rôles

```
SELECT * FROM DBA_ROLES
```

- Créer un rôle

```
CREATE ROLE
```

- Attribuer un privilège/un rôle

```
GRANT privilège/rôle TO rôle/utilisateur
```

- Supprimer un rôle

```
DROP ROLE role
```

## Les bases de données

### MySQL

- Liste des bases de données :

```
SHOW DATABASES;
```

### PostgreSQL

- Liste des bases de données: (Plus d'infos <sup>[15]</sup>)

```
SELECT datname FROM pg_database;  
en général on utilise la métacommande \l
```

**Oracle**

```
SELECT name FROM V$DATABASE
```

**Tables****MySQL**

- Liste des tables

```
SHOW TABLES;
```

**PostgreSQL**

- liste des tables: (Plus d'infos <sup>[16]</sup>)

```
SELECT tablename FROM pg_tables;  
ou la métacommande \dt
```

**Oracle**

- liste des tables:

```
SELECT owner,table_name FROM all_tables;
```

**Schéma des relations entre les tables**

Il est possible de générer un schéma des relations entre les différentes tables grâce à OpenOffice Base.

**MySQL**

Attention, il faut utiliser le moteur InnoDB pour pouvoir ajouter des clés étrangères.

- Lister les clés étrangères

```
SELECT * FROM information_schema.KEY_COLUMN_USAGE T;
```

**PostgreSQL**

- Liste des clefs étrangères:

```
SELECT classid, refclassid FROM pg_depend; (Plus d'infos [17])
```

**Oracle**

- Liste des clés étrangères de l'utilisateur connecté

```
SELECT table_name, constraint_name, r_owner, r_constraint_name  
FROM user_constraints  
WHERE constraint_type = 'R'  
ORDER BY table_name, constraint_name;
```

## Charge de la base de données

Connaître la charge moyenne d'une base de données permet d'anticiper la montée en charge et la migration de la base de données vers un nouveau serveur ou un cluster de serveurs.

Le logiciel Nagios fournit une interface web afin d'afficher un état détaillé du serveur. Il permet également d'envoyer une alerte mail ou SMS à l'administrateur système en cas de panne ou de saturation du serveur.

### MySQL

- MySQLAR est un logiciel qui génère des courbes à partir de l'activité du serveur MySQL, très utile pour auditer des engorgements dans le temps. (Site officiel <sup>[18]</sup> | Démonstration <sup>[19]</sup>)
- Cette commande :

```
# /usr/bin/mysqladmin status
```

permet de connaître la charge du serveur MySQL. Exemple de résultat :

```
/usr/bin/mysqladmin Ver 8.41 Distrib 5.0.18, for pc-linux-gnu on i486  
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX  
DataKonsult AB  
This software comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to modify and redistribute it under the GPL license
```

```
Server version          5.0.18-Debian_7-log  
Protocol version       10  
Connection              Localhost via UNIX socket  
UNIX socket             /var/run/mysqld/mysqld.sock  
Uptime:                 4 min 47 sec
```

```
Threads: 1 Questions: 157 Slow queries: 0 Opens: 0 Flush tables: 1  
Open tables: 64 Queries per second avg: 0.547
```

### Oracle

En version 10g, via la DBConsole.

Sinon, interroger la vue V\$SESSION\_WAIT, qui renseigne en temps réel des sessions en attentes.

Event: correspond à l'événement P1, P2, et P3 correspondent aux spécifications de l'événement (exemple: db file sequential read: P1: file #, P2: block #)

Néanmoins le suivi de l'activité d'une instance Oracle est complexe, et requiert l'utilisation d'outil type PERFSTAT pour avoir des informations concrètes.



## Triggers

Pour afficher les différents triggers définis dans la base de données :

### MySQL

```
SHOW TRIGGERS;
```

### PostgreSQL

- liste des triggers : (Plus d'infos <sup>[20]</sup>)

```
SELECT tgname FROM pg_trigger
```

### Oracle

```
SELECT trigger_name FROM user_triggers;
```

## Procédures stockées

Permet d'afficher la liste des procédures stockées.

### MySQL

```
USE mysql  
SELECT * FROM `func`;
```

(Documentation sur les procédures stockées <sup>[21]</sup>)

### PostgreSQL

Ou utilise la métacommande \df, si on connaît le schéma on peut faire \df schema.\*, par exemple \df public.\* donne toutes les fonctions créées dans ce schéma, qui est celui affecté par défaut à utilisateur. On pourrait tout aussi bien faire \df public.ab\* pour avoir toutes les fonctions commençant par ab. Avec \df+ on obtient aussi les commentaires (CREATE COMMENT) sur la fonction

### Oracle

Pour sélectionner la liste des procédures stockées:

```
SELECT object_name  
FROM user_objects  
WHERE object_type='PROCEDURE';
```

Pour afficher le contenu d'une procédure stockée:

```
SELECT text  
FROM user_source  
WHERE NAME = 'NOM DE LA PROCEDURE'  
AND TYPE = 'PROCEDURE';
```

*On peut aussi utiliser les vues dba\_objects et dba\_source*

## Logs

Les logs sont des fichiers textes qui contiennent tout ce qui se passe dans le logiciel. Ces fichiers sont importants pour détecter et localiser une panne.

Les logs des distributions linux sont dans le dossier : /var/log Le fichier de log le plus important est : /var/log/syslog

## MySQL

- /var/log/mysql.log contient toutes les erreurs autour du serveur MySQL. C'est le premier fichier à regarder quand le serveur ne MySQL démarre pas.

Il est possible d'activer des logs pour sauvegarder toutes les requêtes, les modifications dans les tables ou pour identifier les requêtes les plus lentes. (Détails sur les types de logs [22])

## PostgreSQL

Lors de la création d'une base , elle est configuré avec un archivage minimum (un certains nombre de fichiers de taille fixes sont utilisés pour l'archivage) Cet archivage n'est pas suffisant pour pouvoir restaurer des sauvegardes à chaud, il faut un archivage de type WAL, qui n'écrase pas les log les plus anciens.

Pour configurer un archivage WAL: Dans le fichier postgresql.conf, recherchez le parametre archive\_command et donnez lui la valeur suivante:

```
archive_command = 'cp -i %p /mnt/server/archivedir/%f </dev/null'
(%p correspond au path du fichier à archiver, alors que %f corespond
seulement au nom du fichier)
```

Sous certaines plateformes, la commande cp -i écrase les fichiers déjà existant, ce qui est contraire à la logique d'un archivage WAL. Si vous rencontrez des problemes, il faut rajouter un test dans la commande:

```
archive_command = 'test ! -f ../%f && cp %p ../%f'
```

## Oracle

On distingue deux types de "logs" avec Oracle: - les fichiers "redo logs", qui assurent l'intégrité de la base - les fichiers textes statuant de l'état de la base

- Pour avoir des fichiers "redo log" actifs, la base doit être en mode ARCHIVELOG. Pour connaître le mode d'archivage:

```
SELECT name, log_mode FROM V$DATABASE
```

Pour activer le mode ARCHIVELOG, il faut arrêter la base

```
shutdown immediate
```

puis modifier le fichier init<SID>.ora

```
log_archive_start = TRUE
```

- Les fichiers de log sont également écrits dans le fichier init<SID>.ora

## Exemple :

```
log_archive_dest_1 = "location=/home/oracle/oradata/isima/archive"
log_archive_dest_2 = "location=/home/oracle/temp"
```

On trouvera les fichiers de "log" texte en cherchant les paramètres:

```
SQL> show parameter dump SQL> NAME TYPE VALUE
```

```
-----
----- background_core_dump string partial background_dump_dest
string      /u01/oracle/admin/V102/bdump      core_dump_dest      string
/u01/oracle/admin/V102/cdump max_dump_file_size string UNLIMITED shadow_core_dump
string partial user_dump_dest string /u01/oracle/admin/V102/udump
```

Sachant que: - bdump: processus systèmes - udump: processus "user": client - cdump: fautes de segmentations (bugs)

## Suivi de sécurité

Il est primordial de suivre les alertes à propos de failles de sécurité qui peuvent être détectés dans l'ensemble des logiciels présents sur le serveur.

Heureusement, avec la plupart des distributions linux, dont Debian, le suivi de sécurité est réalisé pour tous les logiciels contenus dans la distribution.

Il suffit de s'abonner à la liste de diffusion [debian-security-announce](mailto:debian-security-announce@lists.debian.org) <sup>[23]</sup>.

à chaque alerte de sécurité, un mail est envoyé à tous les abonnés pour signaler le problème.

Si l'alerte de sécurité concerne un logiciel présent sur le serveur, il faut le mettre à jour grâce aux commandes :

```
# apt-get update
# apt-get upgrade
```

Bien entendu, il faut que les logiciels installés soient des paquetages debian officiels, sinon cette politique de sécurité est défailante.

### Oracle

Debian n'effectue pas de suivi de sécurité pour Oracle, car c'est un logiciel propriétaire. Il faut donc mettre en place une procédure de sécurité spécifique pour Oracle.

Il est impératif, pour un système en production, d'avoir un support valide auprès d'Oracle.

Le site WEB de la base de connaissances est <http://metalink.oracle.com>

## Sauvegardes

Un serveur de base de données doit être obligatoirement sauvegardé car un serveur sera obligatoirement en panne un jour. Il est donc important de mettre en place une politique de sauvegardes et de les effectuer régulièrement. Une multitude de méthodes sont possibles pour sauvegarder un serveur.

La plus courante est de posséder un serveur de sauvegardes qui automatise et centralise toutes les sauvegardes de l'entreprise, puis de stocker sur bande magnétique ou DVD-R.

BackupPc est un logiciel très utilisé pour sauvegarder un ensemble d'ordinateurs, car il possède une interface web afin de lancer des sauvegardes ou de restaurer des fichiers.

## MySQL

- à chaud :

La procédure recommandée pour sauvegarder est :

```
# mysqldump --tab=/emplacement/du/fichier.sql --opt nom_bdd
```

Ceci génère un fichier SQL de la base de données qui est facile à compresser et à sauvegarder. (Documentation détaillée <sup>[24]</sup>)

- à froid :

Sauvegarder le dossier où MySQL stocke ses bases de données. (Généralement /var/lib/mysql)

## PostgreSQL

Les sauvegardes peuvent se faire de trois manières:

- *SQL DUMP*:

Faire un SQL DUMP signifie créer un fichier texte contenant une suite de requêtes SQL permettant de pouvoir reconstruire toute la base de données. Dans PostgreSQL, nous disposons de la commande **pg\_dump** qui s'utilise de la manière suivante.

```
pg_dump dbname > outfile
```

- *Une sauvegarde au niveau du système de fichier*:

Cette sauvegarde se fait elle aussi à froid, et elle consiste simplement en un archivage de toutes les données de la base à sauvegarder. On peut utiliser une commande comme celle ci:

```
tar -cf backup.tar /usr/local/pgsql/data
```

- *Une sauvegarde à chaud*:

Les sauvegardes à chaud se font alors que la base est en exploitation. Pour pouvoir faire des restaurations correctes, il est nécessaire que la base dispose d'un système d'archivage WAL. Cela permet de faire du point-in-time recovery (PITR): On peut récupérer l'état de la base à n'importe quel instant, du moment qu'il y ait eu un backup de fait

Créer la sauvegarde de la base : Après s'être assuré que l'archivage WAL fonctionne, loggez vous en administrateur, et lancez la commande :

```
SELECT pg_start_backup('label');  
(label corespond au nom que vous voulez donner a cette sauvegarde)
```

Ensuite, effectuez une sauvegarde avec la commande `cpio` ou `tar`, comme dans la sauvegarde au niveau du système de fichier.

Et après cette dernière effectuée, reloggez vous en administrateur et lancez la requête :

```
SELECT pg_stop_backup();
```

## Oracle

Il existe plusieurs façons d'effectuer la sauvegarde d'une base de données.

- *Sauvegarde à froid*

Il s'agit de faire une sauvegarde de la base dans un état donné.

- Formatage de texte sous SQL\*Plus

```
Set feedback off
Set Linesize 200
Set Heading off
Set Pagesize 0
Set TrimsPOOL off
Set Verify off
```

- Écrire les données dans un script sql

```
spool ~/backup.sql
select 'host cp ' || name || ' ~/backup ' from v$datafile;
select 'host cp ' || member || ' ~/backup ' from v$logfile;
select 'host cp ' || name || ' ~/backup ' from v$controlfile;
select 'host cp ' || name || ' ~/backup ' from v$tempfile;
spool off
```

- Arrêter la base

```
shutdown immediate
```

- Effectuer la sauvegarde à proprement parler

```
@~/backup.sql
```

- Redémarrer la base

```
startup
```

### *Sauvegarde à chaud*

Il s'agit de sauvegarder des données alors que la base est en marche. Il est nécessaire que le mode ARCHIVELOG soit activé.

- Il faut tout d'abord identifier le tablespace qui va être sauvegardé.
- Ensuite il faut mettre le tablespace hors ligne

### **Exemple :**

```
ALTER TABLESPACE exple_tablespace OFFLINE NORMAL;
```

- Sauvegarder le fichier concerné (un simple copier/coller)
- Enfin, remettre le tablespace en ligne

### **Exemple :**

```
ALTER TABLESPACE exple_tablespace ONLINE NORMAL;
```

## Restauration

### MySQL

Si la sauvegarde a été effectuée à chaud :

```
# mysql < nom_du_fichier.sql
```

Sinon : Arrêter le serveur, et copier la sauvegarde dans le dossier de travail MySQL, puis démarrer le serveur.

### PostgreSQL

La restauration s'effectue de différente manière, suivante le type de suvegarde utilisée.

- *SQL DUMP*

Pour récupérer la base de données, il faut utiliser la commande suivante:

```
psql dbname < infile
```

- *Sauvegarde au niveau du système de fichier*

Il suffit de lancer la commande:

```
tar -xvf backup.tar /usr/local/pgsql/data
```

- *point-in-time recovery (PITR), restauration d'une sauvegarde à chaud*

Après avoir fermé la base de données et restauré le fichier de sauvegarde comme pour une sauvegarde a froid, créez un fichier de configuration pour la restauration dans le répertoire data. Appelez ce fichier *recovery.conf* et dans ce fichier insérez la commande suivante :

```
restore_command = 'cp /mnt/archivedir/%f %p'
```

### Oracle

- Si la sauvegarde a été effectuée à chaud, utiliser la commande 'recover' après avoir éventuellement remis des fichiers sauvegardés

```
RECOVER [AUTOMATIC] [FROM location]
  {[STANDBY] DATABASE [UNTIL options] [USING BACKUP CONTROLFILE]
  |TABLESPACE {tablespace [, tablespace ...]}
  |DATAFILE {datafilename [,datafilename ...]}
  |STANDBY {TABLESPACE tablespace [,tablespace ...]
  |DATAFILE datafilename [,datafilename ...]} UNTIL CONTROLFILE
  |LOGFILE filename
  |CONTINUE [DEFAULT]
  |CANCEL}
  [PARALLEL clause]
```

- Si la sauvegarde a été effectuée à froid, recopier tous les fichiers de la sauvegarde dans la base après l'avoir au préalable arrêtée.

## Mise à jour

Pour mettre à jour le serveur de base de données, s'il est installé en paquetage, il suffit de faire :

```
# apt-get update
# apt-get upgrade
```

Sinon, bon courage :-)

## Suivi des bugs

Quand on rencontre un problème inexplicable avec un logiciel, le bon réflexe est de faire un tour sur le rapporteur de bugs du projet. Il ne faut pas hésiter à rapporter un bug lorsque on en trouve un, car sinon il a moins de chance d'être corrigé.

### MySQL

Le gestionnaire de bugs de MySQL <sup>[25]</sup>.

### PostgreSQL

- Rapporter les bugs :

Lorsque vous rencontrez un bug dans PostgreSQL, vous pouvez le rapporter à l'organisme qui s'occupe des mises à jour de PostgreSQL. Pour cela, il vous suffit d'envoyer un rapport sur le bug à l'adresse suivante : <pgsql-bugs@postgresql.org> Il est demandé de faire un bref descriptif du bug en tant que sujet du mail.

### Oracle

On utilise le site du support Oracle: <http://metalink.oracle.com>

## Références

- [1] <http://dev.mysql.com/doc/refman/5.0/fr/>
- [2] <http://dev.mysql.com/doc/refman/5.1/en/>
- [3] <http://docs.postgresqlfr.org/8.1/>
- [4] [http://www.oracle-10g.de/oracle\\_10g\\_documentation/](http://www.oracle-10g.de/oracle_10g_documentation/)
- [5] <http://www.mysql.com/products/tools/administrator/>
- [6] <http://pgadmin.org/>
- [7] [http://www.oracle.com/enterprise\\_manager/index.html](http://www.oracle.com/enterprise_manager/index.html)
- [8] <http://dev.mysql.com/doc/refman/5.1/en/show.html>
- [9] <http://dev.mysql.com/doc/refman/5.1/en/user-account-management.html>
- [10] <http://dev.mysql.com/doc/refman/5.1/en/create-user.html>
- [11] <http://dev.mysql.com/doc/refman/5.1/en/drop-user.html>
- [12] <http://docs.postgresqlfr.org/8-1/sql-createrole.html>
- [13] <http://docs.postgresqlfr.org/8-1/sql-droprole.html>
- [14] <http://docs.postgresqlfr.org/8.0/user-manag.html>
- [15] <http://docs.postgresqlfr.org/8-1/catalog-pg-database.html>
- [16] <http://docs.postgresqlfr.org/8-1/view-pg-tables.html>
- [17] <http://docs.postgresqlfr.org/8.1/catalog-pg-depend.html>
- [18] <http://gert.sos.be/en/projects/mysqlar/>
- [19] <http://gert.sos.be/demo/mysqlar/>
- [20] <http://docs.postgresqlfr.org/8.2/catalog-pg-trigger.html>
- [21] <http://dev.mysql.com/doc/refman/5.1/en/stored-procedures.html>
- [22] <http://dev.mysql.com/doc/refman/5.1/en/log-files.html>

[23] <http://lists.debian.org/debian-security-announce/>

[24] <http://dev.mysql.com/doc/refman/5.1/en/backup.html>

[25] <http://bugs.mysql.com/>



# Article Sources and Contributors

**Découverte de MySQL, PostgreSQL et Oracle** *Source:* <http://fr.wikibooks.org/w/index.php?oldid=234863> *Contributeurs:* BORIES René, Damja, Dany11, GML, Greudin, Malta, MrFahrenheit, Pitibalrog, Sub, Tavernier, Walpole, Zouff, 105 modifications anonymes

## Licence

---

Creative Commons Attribution-Share Alike 3.0 Unported  
<http://creativecommons.org/licenses/by-sa/3.0/>

---