

MySQL

This FAQ page collects operational notes for **MySQL**.

MySQL replication is broken with "Relay log read failure: ...", how can I fix it?

MySQL multimaster replication needs some advanced MySQL skills. If you do not have them, it is better if you refer to MiRTA PBX support and subscribe a "Nagios monitoring and OS management" service. If instead you have the needed knowledge, understanding that a wrong action can get things worst and you may lose data, to recover from this problem you need:

- Get the MySQL slave status with "show slave status"
- Stop the slave replication with "stop slave"
- Reset the relay log with "reset slave"
- Reinitialize the relay log with "change master to master_log_file='<Relay_Master_Log_File>', master_log_pos=<Exec_Master_Log_Pos>"
- Start the slave replication with "start slave"

I have lost the me_media table

It is easy to panic and to abruptly remove the big voipmonitor.me_media table who is filling your disk, but now you need to recreate it and maybe reduce the retention period for voipmonitor data file. I have created a script as protected/the related application page doing this job.

MySQL time is wrong

It is possible your mysql has no more valid timezone informations. Please run the following command:

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql
```

I need to skip lots of voipmonitor add partition collisions

Why don't use a script like

```
while [ 1 ]; do if [ `echo "show slave status for channel 'from_voip3_to_voip1'" | mysql -u root -ppassw0rd | grep voipmonitor | wc -l` -gt 0 ]; then echo "stop slave for channel 'from_voip3_to_voip1'; set global sql_slave_skip_counter=1; start slave for channel 'from_voip3_to_voip1'" | mysql -u root -ppassw0rd; fi; sleep 5s; echo "show slave status for channel 'from_voip3_to_voip1'\G" | mysql -u root -ppassw0rd | grep Seconds_Behind_Master; done
```

MySQL max open file limit is 5000

The standard MySQL install has an open file limit of 5000 files. To raise this limit, edit the file `/usr/lib/systemd/system/mysqld.service` and set like

```
LimitNOFILE = 32768
```

then reload the service with:

```
systemctl daemon-reload
service mysqld restart
```

If instead you still run CentOS 6, you can add the following to `my.cnf` and restart `mysqld`

```
open_files_limit = 32768
```

I need to connect to MySQL using SSL

From version 4.0.13 is possible to connect to MySQL using SSL. A small change is needed on both `/var/www/html/pbx/include/db.the` related application page and `/var/lib/asterisk/agi-bin/include/db.the` related application page adding the following rows in the right place:

```
$dbconn=NewADOConnection($dbtype);
$dbconn->ssl_capath="";
$dbconn->clientFlags=MYSQLI_CLIENT_SSL;
$GLOBALS['ADODB_SESS_CONN']=$dbconn;
```

I need to load timezone data in MySQL

You can do from your database server running:

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p MySQL
```

Where do I find MySQL 5.5 ?

Use this command:

```
yum install http://repo.mysql.com/yum/mysql-5.5-community/el/7/x86_64/mysql-community-release-el7-5.noarch.rpm
```

How fast should be MySQL?

It is not easy to measure the performance of MySQL, but there is a script you can try to run in `/var/www/html/pbx/protected`

These are some performance analysis, just to compare

```
# php the related application page
Checking voicemail table
Collected 1269 queries
Running 100 cycles
Time required: 11.420516967773
Time for 1000 requests: 8.9996193599475
Checking sipfriends table
Collected 256 queries
Running 100 cycles
Time required: 3.5946228504181
Time for 1000 requests: 14.041495509446
Checking st_states table
Loading 100000 queries
Time required: 14.124006986618
Time for 1000 requests: 0.14124006986618
```

I can't upgrade MySQL, the GPG key is expired or invalid

On CentOS10, I had to do this:

```
rpm -e gpg-pubkey-a8d3785c-6536acda
curl -fsSL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0xB7B3B788A8D3785C" -o
/tmp/fresh.asc
gpg --import /tmp/fresh.asc
\rm /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2023
gpg --output /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2023 --export
BCA43417C3B485DD128EC6D4B7B3B788A8D3785C
```

I get an error "ERROR 1118 (42000) at line 25: Row size too large (> 8126). Changing some columns to TEXT or BLOB may help. In current row format, BLOB prefix of 0 bytes is stored inline."

That because MySQL is exaggerating his needs. Disable it with

```
SET GLOBAL innodb_strict_mode = OFF;
```

Revision #6

Created 2026-06-02 21:58:45 UTC by Admin

Updated 2026-06-05 17:40:40 UTC by Admin