

Database Tips

Some common tricks for quick database optimizations

See what is slowing things down -

```
`show full processlist;
```

Creating Index in old databases

Sometimes adding index will give error when moving from 5.7 to 8 to change that you can set the `sql_mode`

```
SELECT @@SESSION.sql_mode; // See what is the current mode
// For ex for Date error
SET SESSION sql_mode =
' ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER, NO_ENGI
NE_SUBSTITUTION' ;
ALTER TABLE `wp_posts` ADD INDEX `post_date_gmt` (`post_date_gmt`);
```

Upgrading to InnoDB

Quick upgrade to InnoDB

```
SET @DATABASE_NAME = 'name_of_your_db';

SELECT CONCAT(' ALTER TABLE `', table_name, '` ENGINE=InnoDB;') AS sql_statements
FROM information_schema.tables AS tb
WHERE table_schema = @DATABASE_NAME
AND `ENGINE` = 'MyISAM'
AND `TABLE_TYPE` = 'BASE TABLE'
ORDER BY table_name DESC;
```

You will get the queries for all the databases , execute them to convert the tables. If an error regarding the data then set `sql_mode` . That should resolve the issue .

Manually Creating Wordpress Database and login

To list the users -

```
SELECT User, Host FROM mysql.user;
```

log into mysql and -

```
CREATE DATABASE d_{name};  
CREATE USER u_{name}@localhost IDENTIFIED BY '{password}';  
GRANT ALL ON d_{name}.* TO u_{name}@localhost;  
flush privileges;
```

sometimes grant will require password (older versions)

```
GRANT ALL ON d_{name}.* TO u_{name}@localhost identified by '{password}';  
  
GRANT ALL PRIVILEGES ON d_{name}.* TO u_{name}@localhost WITH GRANT OPTION;
```

Manual setting up wordpress database

```
wp core install --title="Site Title" --url={domain_name} --admin_user=siteadmin --  
admin_password={password} --admin_email={email} --skip-email --allow-root
```

Flush Tables issue

Run command from /srv/www

```
find -maxdepth 3 -name "wp-config.php" -exec cat {} \; | grep DB_USER
```

Search replace using -

```
(u_[A-Z0-9]+)  
GRANT RELOAD ON *.* TO '\1'@localhost;
```

Run it inside mysql and it should work now . GRANT reload can only be global.

Importing databases

Import Single Database

```
mysql -uroot -D d_J2MK3T -o d_J2MK3T < db_dump.sql
```

Install mssql client

<https://docs.microsoft.com/en-us/sql/connect/php/installation-tutorial-linux-mac?view=sql-server-ver15> <https://docs.microsoft.com/en-us/sql/connect/php/microsoft-php-drivers-for-sql-server-support-matrix?view=sql-server-ver15#supported-operating-systems>

Get the base ready

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list | sudo tee
/etc/apt/sources.list.d/msprod.list
apt-get update
apt-get install mssql-tools unixodbc-dev
```

Now lets install PHP modules

```
apt install php-pear
apt install php7.4-dev
pecl install sqlsrv
pecl install pdo_sqlsrv
# Create the file sqlsrv.ini and pdo_sqlsrv.ini in modsavailable

# Create links in config

ln -s /etc/php/7.4/mods-available/pdo_sqlsrv.ini 20-pdo_sqlsrv.ini
ln -s /etc/php/7.4/mods-available/sqlsrv.ini 20-sqlsrv.ini
```

SSL Bug : <https://github.com/microsoft/msphpsql/issues/1112#issuecomment-609972220>

Database Recovery

This is to be done if the innodb mysql crashes. First try to do things without putting the DB in recovery mode. If it does not work move to steps below.

Stop Mysql , to copy its files

```
service stop mysql
cp -rf /var/lib/mysql /var/lib/mysql.orig
```

Add **innodb_force_recovery = 1** in my.cnf under [mysqld] , you can increase value from 1 - 6 where mysql starts to work.

Dump all database names -

```
mkdir ~/recovery/
mysql -e 'show databases;' | grep -v information_schema | grep -v Database >
~/recovery/database_list.txt
```

Next dump all databases -

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqldump --skip-lock-tables $db >
/mnt/temp_storage/backup/$db.sql; done
```

Now we drop the databases -

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqladmin drop $db ; done
```

Incase Database tale does not drop , please drop it manually , also move teh ibdata files

```
cd /var/lib/mysql
rm -rf database_name
mv /var/lib/mysql/ibdata1 /mnt/temp_storage/backup/
mv /var/lib/mysql/ib_logfile0 /mnt/temp_storage/backup/
mv /var/lib/mysql/ib_logfile1 /mnt/temp_storage/backup/
```

Remove **innodb_force_recovery = 1** from my.cnf

Start mysql and create the datrabases

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqladmin create $db ; done
```

Import all the databases

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqldump $db <
/mnt/temp_storage/backup/$db.sql; done
```

Repair Databases if any

```
mysqlcheck --repair --all-databases
```

The above should fix the issue hopefully .

Database Recovery 2

This method is the last resort , Incase the databale is very large and the mysql dump keeps failing. We will manually delete the databases.

innodb_force_recovery = 6

This is readonly mode mainly for dumping and not command which mae changes will work.

Step 1 is to run mysql check and see which databases it is faling.

```
service mysql stop
mysqlcheck -c -u root -p --all-databases
```

Mae a note of all databases which have errors , no move those databases out of the /var/lib/mysql directory .

```
mv d_AHDKIT ~/db_backup
mkdir d_AHDKIT
chown -R mysql:mysql d_AHDKIT
```

Now try starting mysql after removing innodb_force_recovery

```
service mysql start
```

Now drop the database with the errors.

```
DROP DATABASE d_AHDKIT;
```

Now you can recreate the database , grant priviliges to use and import from backup.

Revision #15

Created 2 July 2021 05:11:00 by Vikas

Updated 31 January 2023 07:57:37 by Vikas