Restore your Database

In the previous Tutorial we covered how to perform a Database Backup. In this Tutorial we will discuss how to restore the database if you need to. Some reasons for restoring a database would be:

- database became corrupt <
- database inadvertently deleted <
- copying a database from one installation to another <

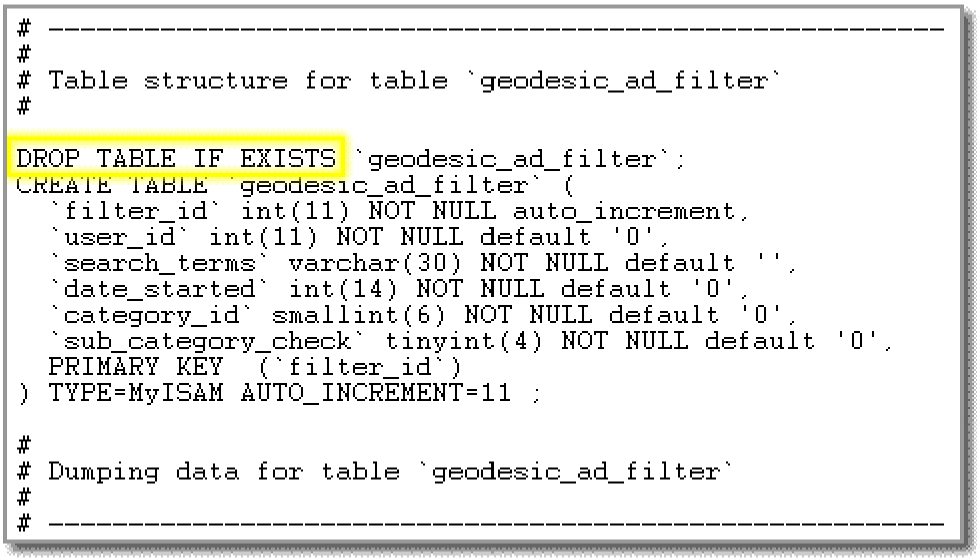
Regardless of the situation, the process is always the same. Keep in mind, however, that when you restore a database you are effectively overwriting the existing database. Therefore, any changes to settings and templates, registrations that took place, listings that were entered, etc. between the time you performed your the back up until now will be lost. With that understanding, proceed as follows:

1. Before going any further you need to first prepare your backup file for importing. What this means is that you need to open up that .sql file and break it apart into smaller files. Why? Because the one single .sql file that you have now is more than likely too large for the server to handle all at one time. In other words, PHP on your server would "time out" before it finished running the entire file, leaving you with only a partially restored database. You can overcome this server limitation by breaking up the single file into several smaller files.

Ideally, you need to break the sql into 5 or 6 separate files. Simply open your .sql file and cut/paste a portion of the code into a new file. Save it and then repeat the process until you have created 5 or 6 separate files from the original .sql file. Your new files can be saved as any name you like. However, we recommend naming them by number so that you can maintain the order in which you upload them in a later step. ORDER IS IMPORTANT. For example, when you are finished you should have a set a files that look something like this: 1.sql, 2.sql, 3.sql, 4.sql, 5.sql, 6.sql

As long as you keep the files in order as described above, you don't have to worry about splitting up a large sql table between two files. If PHP finishes running a partial table in one file, it will then pick right back up and finish that table in the next file, if you upload them in the proper order.

<tip c w>**Important:** Ensure that your .sql file contains DROP TABLES for each table within the code. Failure to do so will result in error messages being displayed in phpMyAdmin. If you followed the steps in the "Backing up Your Database" Tutorial, then your .sql file should have drop tables in the code. See example below of what the first line in each table should look like.</tip>



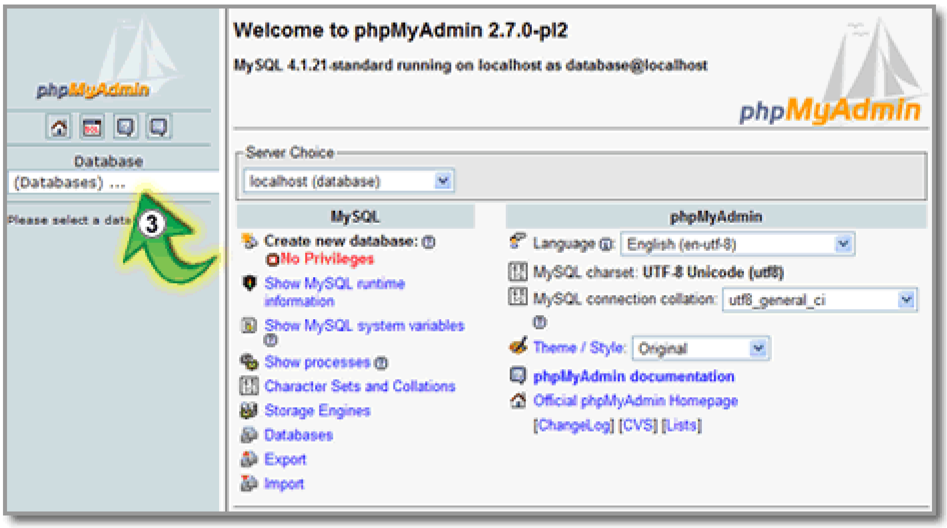
```
# --------------------------------------------------------
#
# Table structure for table `geodesic_ad_filter`
#

DROP TABLE IF EXISTS `geodesic_ad_filter`;
CREATE TABLE `geodesic_ad_filter` (
  `filter_id` int(11) NOT NULL auto_increment,
  `user_id` int(11) NOT NULL default '0',
  `search_terms` varchar(30) NOT NULL default '',
  `date_started` int(14) NOT NULL default '0',
  `category_id` smallint(6) NOT NULL default '0',
  `sub_category_check` tinyint(4) NOT NULL default '0',
  PRIMARY KEY  (`filter_id`)
) TYPE=MyISAM AUTO_INCREMENT=11 ;

#
# Dumping data for table `geodesic_ad_filter`
#
# --------------------------------------------------------
```

<

1. Log in to phpMyAdmin.

<tip c n>**Note:** The page layout and required steps may vary slightly depending upon the version of phpMyAdmin that is installed on your domain.</tip> <

1. Select the database name you want to restore from the dropdown.



Welcome to phpMyAdmin 2.7.0-pl2

MySQL 4.1.21-standard running on localhost as database@localhost

Server Choice: localhost (database)

Database: (Databases) ...

Please select a data

Create new database: No Privileges

Language: English (en-utf-8)

MySQL charset: UTF-8 Unicode (utf8)

MySQL connection collation: utf8_general_ci

Theme / Style: Original

phpMyAdmin
3
MySQL
Show MySQL runtime information
Show MySQL system variables
Show processes
Character Sets and Collations
Storage Engines
Databases
Export
Import
phpMyAdmin
phpMyAdmin documentation
Official phpMyAdmin Homepage
[ChangeLog] [CVS] [Lists]

<

1. When the screen refreshes you will see a list of tables that are currently present within the database you just selected. The same set of table names will display on both sides of the screen. (If this is a new database, or if you have previously 'dropped' the database tables, you will not see any.) You will see several tabs at the top of the screen. Click on the "Import" tab.

<tip c n>**Note:** Older versions of phpMyAdmin will not have an "import" tab. Instead, you will need to click on the "SQL" tab, as it will be your only option for importing SQL.</tip>

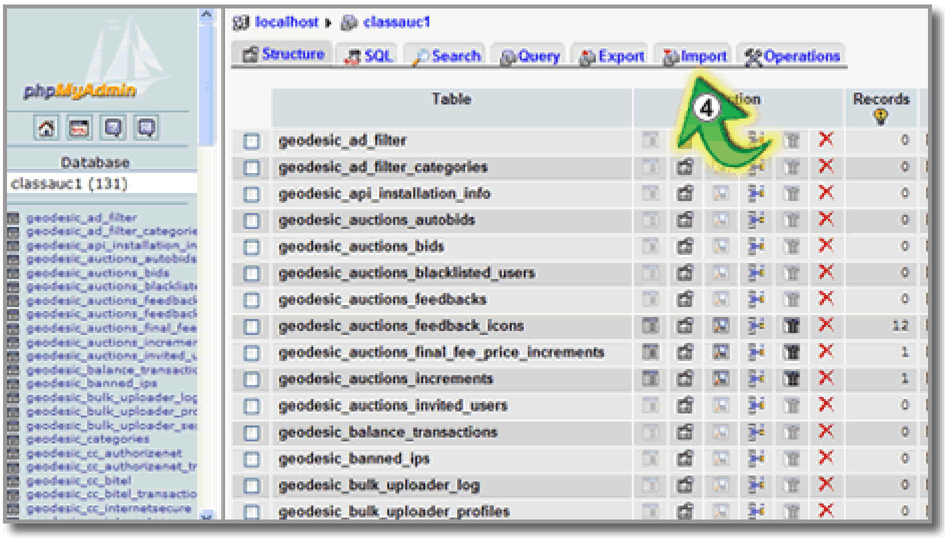
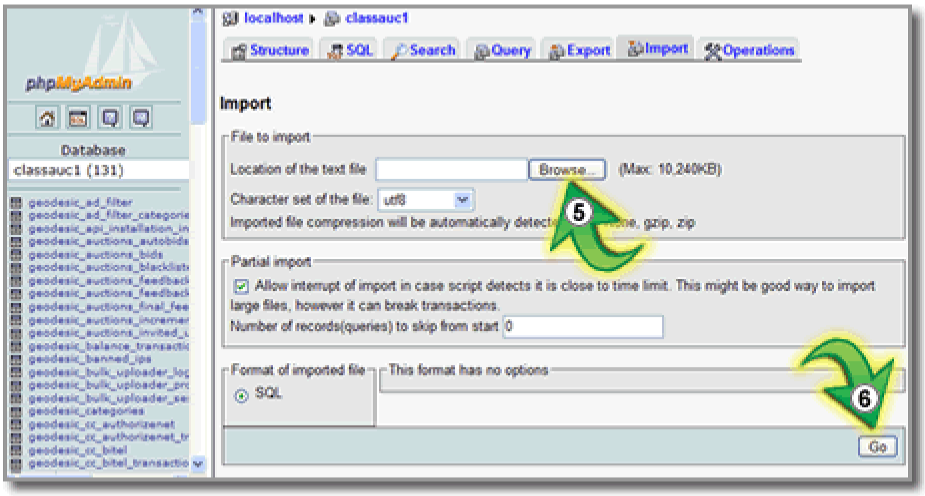


	Table	Records
□	geodesic_ad_filter	0
□	geodesic_ad_filter_categories	0
□	geodesic_api_installation_info	0
□	geodesic_auctions_autobids	0
□	geodesic_auctions_bids	0
□	geodesic_auctions_blacklisted_users	0
□	geodesic_auctions_feedbacks	0
□	geodesic_auctions_feedback_icons	12
□	geodesic_auctions_final_fee_price_increments	1
□	geodesic_auctions_increments	1
□	geodesic_auctions_invited_users	0
□	geodesic_balance_transactions	0
□	geodesic_banned_ips	0
□	geodesic_bulk_uploader_log	0
□	geodesic_bulk_uploader_profiles	0

<

1. You should now see a screen that looks similar to the image below. Click on the "Browse" button (#5) and upload the first your 5 or 6 .sql files you created in step 1 above using the popup window that appears. Once you have selected the file, click the "Go" button (#6) to upload the file. In a few moments you should see the screen refresh and the first set of SQL tables will be entered into the database. Repeat this process until you have uploaded all of your .sql files in order.



Import

File to import

Location of the text file (Max: 10,240KB)

Character set of the file: utf8

Imported file compression will be automatically detect… …ne, gzip, zip

Partial import

☑ Allow interrupt of import in case script detects it is close to time limit. This might be good way to import large files, however it can break transactions.

Number of records(queries) to skip from start 0

Format of imported file

● SQL

This format has no options

<

1. You should now be able to log in to the software's admin panel using your admin username and password. If you are unable to, you can reset the admin username and password as described in the "Reset Admin Password" tutorial.

<tip c w>**IMPORTANT:** Ensure that your .sql file contains DROP TABLES for each table within the code. Failure to do so will result in error messages being displayed in phpMyAdmin. If you followed the steps in the "Backing up Your Database" Tutorial, then your .sql file should have drop tables in the code. See example below of what the first line in each table should look like.</tip> <

Working Through Problems When Running Queries

We have had a bit of experience importing databases and can provide a few pointers.

Break your export into several "bite size" pieces so that each file doesn't exceed the "max file size" displayed within the "file import" feature of the Phpmyadmin tool you are using. Then make sure to import these files in order as they were in the one original file. There could have been a "table create" statement created in a previous file allowing subsequent files to insert into properly.

The geodesic_classifieds table is the the hardest to import. It is usually the largest table especially on larger sites but the main problem is that table contains user inserted content. Almost all problems we encounter in this table are in the description field of this table. What we have done in the past is run a file (as broken up as suggested above). If you see an error go to the database table currently in your database and find the last sql query run. Basically go to the end of the table and see what the last listing id is (this is the "id" field) in that table. Return to your import file and go to the next insert query that failed. Hopefully these are sequential by the "id" field. At this point you can do a couple things.

You can remove this query from this import file and place it within a specific place to be run later individually. This may be the best way to get through the bulk of the import. Depends on what you feel for your situation but our experience shows there to be few queries with a problem. If there are many queries like this you may just need to stop and deal now with the error or stopping every few queries can cost more time. If you decide to set it aside for future debugging set aside this query (cut and paste out of this file) in a "to do later" file. You may need to add other queries to this file as you work through the import. Once this query has been removed from the import file you will need to remove all previously successfully run queries in the same import file. If you try these same queries again the database will not be corrupted in most cases but you will see an "already exists in table" error. The previous holds true in the case of tables that have keys/primary keys where there can be no duplicates. If the table you are working on does not have a unique key field within it you may need to delete that specific table and retract in the import file to the point that table creation statement and start importing from that point on (removing all queries before the table creation statement needing to be run).

You can also choose to deal with the query error at this point before going further into the import. You will certainly have to deal with the error now if the specific query causing the problem is a "table creation" statement. Run the problem query individually. You usually can see better where the problem is if the query is run individually. Most of the problems happen in user inserted content. At this point you may need to make a judgment call. Many times just changing the user submitted content a bit so that the meaning/intent is still there is what's needed. You may also do a search of the problem string through the queries not already run to see if you can do a simple search and replace to fix the problem in many queries. Hopefully the problem you see is unique.

If you stopped the import process to deal with a specific sql query make sure to remove all "already run" queries before you run the same import file before you start the import process again.

You may not decide to run the query at all. You have to do a little investigation and possibly a question or two to support. We suggest you run all sql queries to make sure you have a complete duplicate of the state of the database at the time the backup was taken. But if the query you are trying to run if for a listing (geodesic_classifieds table) that is about to be archived (at 30 days expired by default) or a listing that has been archived (geodesic_classifieds_expired table) for years. This is a judgment call we suggest that you question support about.