

Restore dumped MySQL data on CentOS 7

Restoring dumped MySQL data is something painful. `_restore_mysql.sh` enables you to restore the dumped MySQL data easily.

You can simply restore the dumped MySQL data as following:

```
_restore_mysql.sh example.sql
```

Above will create "example" as a database, and restore all the content.

If you have multiple dump files, you can do it as following:

```
_restore_mysql.sh example1.sql example2.sql
```

Below is another option:

```
_restore_mysql.sh *.sql
```

Below is the whole source code can run on your server. It requires PHP at least.

_restore_mysql.sh

```
#!/usr/bin/php
<?php
/*
 * Restre dumped MySQL Data
 *
 * Author: Chun Kang (ck@ckii.com)
 * Date: 2021-10-22
 *
 **/

$db = [
    "host" => "127.0.0.1",
    "port" => 3306,
    "user_id" => NULL, // default user ID
    "password" => NULL // DB Password
];

echo "Please input MySQL user ID (ENTER to use the default value):\n";
$tmp = trim (readline());
if (strlen($tmp)) $db['user_id'] = $tmp;

echo "Please input MySQL user password (ENTER to use the default value):\n";
$tmp = trim (readline());
if (strlen($tmp)) $db['password'] = $tmp;

echo "\n";

// connect to MySQL Server
$pdo = new PDO("mysql:host={$db['host']};port={$db['port']}", $db['user_id'], $db['password']);

foreach($argv as $backup_filename)
{
    if (!preg_match("/\.sql$/", $backup_filename)) continue;

    $pathinfo = pathinfo( $backup_filename);

    $database_name = $pathinfo['filename'];

    // drop database
    $pdo->query("DROP DATABASE {$database_name}");

    // drop database
    $pdo->query("CREATE DATABASE {$database_name} CHARACTER SET utf8mb4 COLLATE utf8mb4_bin");

    echo "Processing ... {$database_name}\n";
    $cmd = "mysql -u{$db['user_id']} -p{$db['password']} {$database_name} < {$backup_filename}";
    shell_exec( $cmd);

    echo "\n\n";
}

echo "All the processing is done.\n\n";
```