

**mysqldump** is an effective tool to backup MySQL database. It creates a \*.sql file with **DROP table**, **CREATE table** and **INSERT into** sql-statements of the source database. To restore the database, execute the \*.sql file on destination database. For MyISAM, use [mysqlhotcopy method](#) that we explained earlier, as it is faster for MyISAM tables.

Using mysqldump, you can backup a local database and restore it on a remote database at the same time, using a single command. In this article, let us review several practical examples on how to use mysqldump to backup and restore.

For the impatient, here is the quick snippet of how backup and restore MySQL database using mysqldump:

**backup:** # mysqldump -u root -p[root\_password] [database\_name] > dumpfilename.sql

**restore:** # mysql -u root -p[root\_password] [database\_name] < dumpfilename.sql

## How To Backup MySQL database

### 1. Backup a single database:

This example takes a backup of sugarcrm database and dumps the output to sugarcrm.sql

```
# mysqldump -u root -ptmppassword sugarcrm > sugarcrm.sql
```

```
# mysqldump -u root -p[root_password] [database_name] > dumpfilename.sql
```

The sugarcrm.sql will contain drop table, create table and insert command for all the tables in the sugarcrm database. Following is a partial output of sugarcrm.sql, showing the dump information of accounts\_contacts table:

```
--
-- Table structure for table `accounts_contacts`
--

DROP TABLE IF EXISTS `accounts_contacts`;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `accounts_contacts` (
  `id` varchar(36) NOT NULL,
  `contact_id` varchar(36) default NULL,
  `account_id` varchar(36) default NULL,
  `date_modified` datetime default NULL,
  `deleted` tinyint(1) NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `idx_account_contact` (`account_id`,`contact_id`),
  KEY `idx_contid_del_accid` (`contact_id`,`deleted`,`account_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
SET character_set_client = @saved_cs_client;

--
-- Dumping data for table `accounts_contacts`
--

LOCK TABLES `accounts_contacts` WRITE;
/*!40000 ALTER TABLE `accounts_contacts` DISABLE KEYS */;
INSERT INTO `accounts_contacts` VALUES ('6ff90374-26d1-5fd8-b844-4873b2e42091',
'11ba0239-c7cf-e87e-e266-4873b218a3f9','503a06a8-0650-6fdd-22ae-4873b245ae53',
'2008-07-23 05:24:30',1),
```

```
( '83126e77-eeda-f335-dc1b-4873bc805541', '7c525b1c-8a11-d803-94a5-4873bc4ff7d2',
'80a6add6-81ed-0266-6db5-4873bc54bfb5', '2008-07-23 05:24:30', 1),
( '4e800b97-c09f-7896-d3d7-48751d81d5ee', 'f241c222-b91a-d7a9-f355-48751d6bc0f9',
'27060688-1f44-9f10-bdc4-48751db40009', '2008-07-23 05:24:30', 1),
( 'c94917ea-3664-8430-e003-487be0817f41', 'c564b7f3-2923-30b5-4861-487be0f70cb3',
'c71eff65-b76b-cbb0-d31a-487be06e4e0b', '2008-07-23 05:24:30', 1),
( '7dab11e1-64d3-ea6a-c62c-487ce17e4e41', '79d6f6e5-50e5-9b2b-034b-487ce1dae5af',
'7b886f23-571b-595b-19dd-487ce1eee867', '2008-07-23 05:24:30', 1);
/*!40000 ALTER TABLE `accounts_contacts` ENABLE KEYS */;
UNLOCK TABLES;
```

## 2. Backup multiple databases:

If you want to backup multiple databases, first identify the databases that you want to backup using the show databases as shown below:

```
# mysql -u root -ptmppassword
```

```
mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| bugs                    |
| mysql                   |
| sugarcrm                 |
+-----+
4 rows in set (0.00 sec)
```

For example, if you want to take backup of both sugarcrm and bugs database, execute the mysqldump as shown below:

```
# mysqldump -u root -ptmppassword --databases bugs sugarcrm > bugs_sugarcrm.sql
```

Verify the bugs\_sugarcrm.sql dumpfile contains both the database backup.

```
# grep -i "Current database:" /tmp/bugs_sugarcrm.sql
-- Current Database: `mysql`
-- Current Database: `sugarcrm`
```

## 3. Backup all the databases:

The following example takes a backup of all the database of the MySQL instance.

```
# mysqldump -u root -ptmppassword --all-databases > /tmp/all-database.sql
```

## 4. Backup a specific table:

In this example, we backup only the accounts\_contacts table from sugarcrm database.

```
# mysqldump -u root -ptmppassword sugarcrm accounts_contacts \
> /tmp/sugarcrm_accounts_contacts.sql
```

## 4. Different mysqldump group options:

- ***–opt is a group option***, which is same as ***–add-drop-table***, ***–add-locks***, ***–create-options***, ***–quick***, ***–extended-insert***, ***–lock-tables***, ***–set-charset***, and ***–disable-keys***. ***opt*** is enabled by default, disable with ***–skip-opt***.

- ***–compact is a group option***, which gives less verbose output (useful for debugging). Disables structure comments and header/footer constructs. Enables options `–skip-add-drop-table` `–no-set-names` `–skip-disable-keys` `–skip-add-locks`

## How To Restore MySQL database

### 1. Restore a database

In this example, to restore the sugarcrm database, execute mysql with < as shown below. When you are restoring the dumpfilename.sql on a remote database, make sure to create the sugarcrm database before you can perform the restore.

```
# mysql -u root -ptmppassword
```

```
mysql> create database sugarcrm;  
Query OK, 1 row affected (0.02 sec)
```

```
# mysql -u root -ptmppassword sugarcrm < /tmp/sugarcrm.sql
```

```
# mysql -u root -p[root_password] [database_name] < dumpfilename.sql
```

### 2. Backup a local database and restore to remote server using single command:

This is a sleek option, if you want to keep a read-only database on the remote-server, which is a copy of the master database on local-server. The example below will backup the sugarcrm database on the local-server and restore it as sugarcrm1 database on the remote-server. Please note that you should first create the sugarcrm1 database on the remote-server before executing the following command.

```
[local-server]# mysqldump -u root -ptmppassword sugarcrm | mysql \
```

```
                -u root -ptmppassword --host=remote-server -C sugarcrm1
```

[Note: There are two -- (hyphen) in front of host]