MySQL 数据库灾备的基础知识大全

平时我们在使用 MySQL 数据库的时候经常会因为操作失误造成数据丢失，MySQL 数据库备份可以帮助我们避免由于各种原因造成的数据丢失或着数据库的其他问题。

一、数据备份捷径

因为这个方法没有得到官方正式文档的验证，我们暂称为试验吧。

目的：备份 hostA 主机中一个 MySQL 数据库备份 TestA，并恢复到到 hostB 机中

试验环境：

操作系统：WinNT4.0,Mysql3.22.34,phpMyAdmin 2.1.0

在 hostA 中安装 MySQL 数据库备份并建立 TestA 数据库

hostB 机安装 MySQL 数据库备份，没有 TestA 数据库

方法步骤：

启动 phpMyAdmin 察看 HostA 和 HostB 中的数据库列表，在 HostB 中没有 TestA 数据库，找到 HostA 中 MySQL 数据库备份的安装目录，并找到数据库目录 data，在我的试验环境中，这个目录是 C:\mysql\data，找到对应数据库名称的子目录 C:\mysql\data\TestA，粘贴拷贝到 HostB 的 Data 目录下，是 HostA 同 HostB MySQL 数据库备份数据目录下的文件相同。

刷新 HostB 的 phpMyAdmin 察看一下数据库列表，我们看到 TestA 已经出现，并且作查询修改等操作都正常，备份恢复恢复成功。

试验结论：MySQL 的数据库可以通过文件形式保存，备份，恢复只要将相应文件目录恢复即可，无需使用其它工具备份。

二、正规的方法(官方建议)：

导出要用到 MySQL 数据库备份的 mysqldump 工具，基本用法是：

```
mysqldump [OPTIONS] database [tables]
```

如果你不给定任何表，整个数据库将被导出。

通过执行 mysqldump --help，你能得到你 mysqldump 的版本支持的选项表。

注意，如果你运行 mysqldump 没有--quick 或--opt 选项，mysqldump 将在导出结果前装载整个结果集到内存中，如果你正在导出一个大的数据库，这将可能是一个问题。

mysqldump 支持下列选项：

```
--add-locks
```

在每个表导出之前增加 LOCK TABLES 并且之后 UNLOCK TABLE。(为了使得更快地插入到 MySQL 数据库备份)。

```
--add-drop-table
```

在每个 create 语句之前增加一个 drop table。

```
--allow-keywords
```

允许创建是关键词的列名字。这由在列名前面加表名的方法做到。

```
-c, --complete-insert
```

使用完整的 insert 语句(用列名字)。

```
-C, --compress
```

如果客户和服务器均支持压缩，压缩两者间所有的信息。

```
--delayed
```

用 INSERT DELAYED 命令插入行。

```
-e, --extended-insert
```

使用全新多行 INSERT 语法。(给出更紧缩并且更快的插入语句)

```
-#, --debug[=option_string]
```

跟踪程序的使用(为了调试)。

```
--help
```

显示一条帮助消息并且退出。

```
--fields-terminated-by=……
--fields-enclosed-by=……
--fields-optionally-enclosed-by=……
--fields-escaped-by=……
--fields-terminated-by=……
```

这些选择与-T 选择一起使用，并且有相应的 LOAD DATA INFILE 子句相同的含义。

LOAD DATA INFILE 语法。

```
-F, --flush-logs
```

在开始导出前，洗掉在 MySQL 数据库备份服务器中的日志文件。

```
-f, --force,
```

即使我们在一个表导出期间得到一个 SQL 错误，继续。

```
-h, --host=.
```

从命名的主机上的 MySQL 数据库备份服务器导出数据。缺省主机是 localhost。

```
-l, --lock-tables.
```

为开始导出锁定所有表。

```
-t, --no-create-info
```

不写入表创建信息(CREATE TABLE 语句)

```
-d, --no-data
```

不写入表的任何行信息。如果你只想得到一个表的结构的导出，这是很有用的！

```
    --opt
```

同：

```
    --quick --add-drop-table --add-locks --extended-insert
 --lock-tables
```

应该给你为读入一个 MySQL 数据库备份服务器的尽可能最快的导出。

```
    -pyour_pass, --password[=your_pass]
```

与服务器连接时使用的口令。如果你不指定“=your_pass”部分，mysqldump 需要来自终端的口令。

```
    -P port_num, --port=port_num
```

与一台主机连接时使用的 TCP/IP 端口号。(这用于连接到 localhost 以外的主机，因为它使用 Unix 套接字。)

```
    -q, --quick
```

不缓冲查询，直接导出至 stdout;使用 MySQL_use_result()做它。

```
    -S /path/to/socket, --socket=/path/to/socket
```

与 localhost 连接时(它是缺省主机)使用的套接字文件。

```
    -T, --tab=path-to-some-directory
```

对于每个给定的表，创建一个 table_name.sql 文件，它包含 SQL CREATE 命令，和一个 table_name.txt 文件，它包含数据。 注意：这只有在

mysqldump 运行在 mysqld 守护进程运行的同一台机器上的时候才工作。txt 文件的格式根据--fields-xxx 和--lines--xxx 选项来定。

```
-u user_name, --user=user_name
```

与服务器连接时，MySQL 使用的用户名。缺省值是你的 Unix 登录名。

```
-O var=option, --set-variable var=option
```

设置一个变量的值。可能的变量被列在下面。

```
-v, --verbose
```

冗长模式。打印出程序所做的更多的信息。

```
-V, --version
```

打印版本信息并且退出。

```
-w, --where='where-condition'
```

只导出被选择了的记录;注意引号是强制的！

```
"--where=user='jimf'“ ”-wuserid>1“ ”-wuserid1"
```

最常见的 mysqldump 使用可能制作整个数据库的一个备份：

```
mysqldump --opt database > backup-file.sql
```

但是它对用来自于一个数据库的信息充实另外一个 MySQL 数据库备份也是有用的：

```
mysqldump --opt database
MySQL--host=remote-host -C database
```

由于 mysqldump 导出的是完整的 SQL 语句，所以用 MySQL 数据库备份客户程序很容易就能把数据导入了：

```
mysqladmin create target_db_name
MySQL target_db_name backup-file.sql
```

MySQL 数据库备份虽然大家在平时不经常遇到，但是多熟悉一门知识总不是坏事。MySQL 数据库备份在使用的时候对于刚刚接触的人来说是保护数据库不会由于认为操作失误而导致数据丢失。