



# 数据库技术与应用

---

## 第六章 MySQL备份与恢复





## 本章主要内容

任何数据库系统都会由于各种原因出现错误，甚至崩溃，因此数据库必须要有一套完备的灾难恢复机制。一般通过数据库日志、数据备份、数据恢复相关技术来实现

### √ 6.1 MySQL 日志

- \* 6.1.1 错误日志
- \* 6.1.2 二进制日志
- \* 6.1.3 查询日志
- \* 6.1.4 慢日志

### √ 6.2 MySQL 备份与恢复

- \* 6.2.1 备份/恢复策略
- \* 6.2.2 逻辑备份和恢复
- \* 6.2.3 物理备份和恢复
- \* 6.2.4 表的导入导出





# MYSQL 日志概述

MySQL 日志记录了数据库工作的各种信息，以帮助数据库管理员追踪数据库曾经发生过的各种事件，从而实现对数据库系统的各种维护和优化。MYSQL 日志类型如下表所示：

日志文件	记入文件中的信息类型
错误日志	记录启动、运行或停止mysqld时出现的问题。
二进制日志	记录所有更改、复制数据的语句。如DDL、DML语句，不包括数据查询语句
查询日志	记录建立的客户端连接和执行的所有语句。
慢日志	记录所有执行时间超过long_query_time秒的所有查询或不使用索引的查询。

**MYSQL**安装后默认仅启动错误日志（且不能被禁止），其它日志则需要通过配置服务器来启动。**MYSQL**的日志文件一般放在默认的数据目录中（**datadir**）。默认数据目录通过查看配置文件**my.ini**进行查看。



## 6.1.1 错误日志

MySQL错误日志记录了MySQL服务开启、停止、以及运行过程中出现的错误信息。

### √设置与查看错误日志

- \* **MySQL错误日志默认开启，且不能禁止。**
- \* 错误日志文件通常名称为 hostname.err。
- \* 在my.ini文件的[mysql d]组中加入log-error选项来指定错误文件的存放位置及文件名。

### √删除错误日志

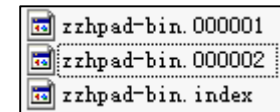
- \* Mysql admin开启新错误文件：`mysqladmin -u root -p flush-logs`
- \* FLUSH LOGS开启新错误文件（必须先登录MySQL服务器）



## 6.1.2 二进制日志

二进制日志（更新日志）记录所有更改、复制数据的语句。如DDL、DML语句，不包括数据查询语句。服务器启动二进制日志性能大约慢1%。

### √启动与设置二进制日志



- \* 在`my.ini`文件的`[mysqld]`组中加入`log-bin[=dir\[filename]]`选项来指定二进制文件存放位置及文件名。
- \* MySQL服务每次重启都会生成一个新的二进制日志文件，文件为`filename.number`，其中`number`逐次递增。
- \* 启动二进制日志后，还会生成一个`filename.index`二进制清单文件。

### √查看二进制日志文件

```
mysql> insert into qq values('345435345',3,'钟老师','zzhzzh@qq.com');  
Query OK, 1 row affected (0.08 sec)
```

```
C:\mysql\mysql15\data>mysqlbinlog zzhpad-bin.000002
```

- \* `Mysqlbinlog filename.number`



## 6.1.2 二进制日志

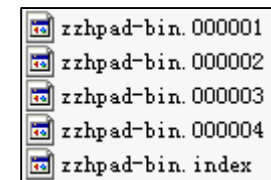
### √ 删除二进制日志

- \* 删除所有二进制日志:

Reset master; 删除后系统自动建立一个序号为000001的新日志

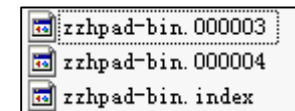
- \* 删除指定number序号前的所有日志文件:

purge master logs to 'zzhpad-bin.000003';



- \* 删除指定时间之前的所有日志文件:

purge master logs before '2011-09-27 10:51:00';



- \* 设置过去天数, 从而自动删除过期日志:

my.ini文件的[mysqld]组中加入 expire\_logs\_days=3





## 6.1.3 查询日志

通用查询日志记录了所有连接和查询语句。查询日志能够全面的反应MySQL服务器中发生的事件，但是查询日志需占用较大的系统资源。

### √启动与设置查询日志

- \* 在`my.ini`文件的`[mysqld]`组中加入`log[=dir\[filename]]`选项来指定二进制文件存放位置及文件名。
- \* 在没有指定`filename`时，默认文件名为`host_name.log`。

### √查看查询日志文件

- \* 查询日志为纯文本文件（例`zzhpad.log`），可以直接打开查看。

### √删除日志文件

- \* `mysqladmin -u root -p flush-logs` 重置覆盖原来日志。
- \* 直接通过文件系统进行删除





## 6.1.4 慢查询日志

慢查询日志记录了所有执行时间超过指定时间（默认为10秒）的SQL查询语句，其中获得表锁定的时间不算作执行时间。可用于找出效率低下的查询，从而进行优化。

### √启动与设置查询日志

- \* my.ini 文件的[mysqld]组中加入如右所示的参数。
- \* 在没指定日志文件名时，默认文件名为host\_name-slow.log。

```
log-slow-queries
long_query_time=10
```

### √查看查询日志文件

- \* 慢日志为纯文本文件（例zzhpad-slow.log），
- \* Mysql dumpslow zzhpad-slow.log 可对慢日志进行分类汇总，从而从众多记录中快速找到感性趣的记录。

```
mysql> show variables like 'long%';
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| long_query_time | 10.000000     |
+-----+-----+
```

```
mysql> set long_query_time=2;
Query OK, 0 rows affected (0.05 sec)

mysql> show variables like 'long%';
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| long_query_time | 2.000000      |
+-----+-----+
```





# MYSQL 日志小结

日志文件	记入文件中的信息类型
错误日志	记录启动、运行或停止mysqld时出现的问题。
二进制日志	记录所有更改、复制数据的语句。如DDL、DML语句，不包括数据查询语句
查询日志	记录建立的客户端连接和执行的所有语句。
慢日志	记录所有执行时间超过long_query_time秒的所有查询或不使用索引的查询。





## 6.2 MYSQL备份与恢复

操作系统崩溃、电源故障、文件系统崩溃、硬件问题(硬盘、主板等等)等问题，会导致数据库服务中断或数据丢失，在此情况下，必须通过一组有效地策略来保证数据库服务器完好如初。

- \* 6.2.1 备份/恢复策略
- \* 6.2.2 逻辑备份和恢复
- \* 6.2.3 物理备份和恢复
- \* 6.2.4 表的导入导出





## 6.2.1 备份/恢复策略

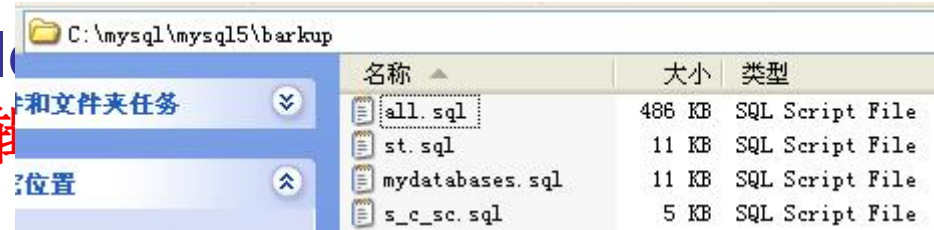
- ✓ 确定要备份表的存储引擎是事务型还是非事务型，不同存储引擎备份方式在数据一致性方面是不同的。
- ✓ 确定使用完全备份还是增量备份。
- ✓ 可以采取复制数据库文件的方法进行异地备份，但复制不能代替备份，复制对数据库的误操作无能为力。
- ✓ 按一定周期定期做备份。备份要在系统负载较小的时候进行。
- ✓ 确保MySQL打开了log-bin选项，有了二进制日志，MySQL才可以在必要的时候做完整恢复，或基于时间点的恢复，或基于位置的恢复。
- ✓ 注意做备份恢复测试，确保备份是有效、且可以恢复的。





## 6.2.2 逻辑备份和恢复

✓ **MYSQL逻辑备份使用mysqldump**  
数据备份为一个文本文件。逻辑



名称	大小	类型
all.sql	486 KB	SQL Script File
st.sql	11 KB	SQL Script File
mydatabases.sql	11 KB	SQL Script File
s_c_sc.sql	5 KB	SQL Script File

- \* `Mysqldump -uroot -p --all-database>all.sql`
- \* `Mysqldump -uroot -p --single-transaction st>st.sql`
- \* `Mysqldump -uroot -p --database st test>mydatabases.sql`
- \* `Mysqldump -uroot -p -l st student course sc>s_c_sc.sql`

**注意：**mysqldump的选项参数非常多，详情可以参考**MYSQL手册**。在此介绍其中三个参数以保证数据一致性：

- lock-tables, -l**：备份时锁定数据表，对于**MyISAM**表而言适用
- single-transaction**：产生一个数据快照，对于**InnoDB**表而言适用
- master-data**：将二进制日志的位置和文件名写入到输出中。对主从结构适用
- flush-logs, -F**：开始转储前刷新**MySQL**服务器日志文件。



## 6.2.2 逻辑备份和恢复

√ 利用mysqldump与二进制日志实现增量恢复。

### \* 一、备份

- `Mysqldump -uroot -p -l -F st>st.sql`
- -l表示锁定INNODB表，-F表示备份时刷新日志（如mylog.000002）

### \* 二、进行各种数据操作...然后重新启动服务器

### \* 三、还原

- 1、`Mysql -uroot -p st<st.sql` 还原备份是数据库状态
- 2、`Mysqlbinlog mylog.000002|mysql -uroot -p st`恢复日志中记录的操作





## 6.2.3 物理备份和恢复

物理备份分为冷备份和热备份两种，与逻辑备份相比，物理备份最大的优点是备份和恢复得速度更快，其原因是物理备份都是基于文件的拷贝。

✓**冷备份**：停止数据库服务，拷贝数据文件。该方法适用于MyISAM和InnoDB引擎。基本步骤：

- \* **第一步 备份**：停止服务，将**MYSQL数据库文件**和日志文件拷贝到备份目录。
- \* **第二步：恢复**：停止服务，将备份文件覆盖当前**MYSQL**数据文件；启动数据服务，适用**mysqlbinlog**工具从二进制日志中恢复自备份以来的所有日志。



## 6.2.3 物理备份和恢复

✓ **热备份**：不必中断数据库服务进行备份。但对于**MyISAM**和**InnoDB**引擎热备份的方法是不同的。

\* **MyISAM**存储引擎：

- **LINUX、UNIX**下可以使用**MYSQL**自带工具**mysqlhotcopy**

\* 利用**MySQL Administrator**的备份与恢复工具：





## 6.2.4 表的导入导出

### √ SELECT ...INTO OUTFILE 导出文本文件基本语法

**Select [列名] from table [where语句] into outfile '文件路径'[OPTION]**

#### OPTION选项:

**FIELDS TERMINATED BY '字符串'**: 设置字段的分隔符, 默认为“\t”

**FIELDS ENCLOSED BY '字符'**: 设置字段值包含在什么字符之中, 默认无字符

**FIELDS OPTIONALLY ENCLOSED BY '字符'**: 设置字符串类型字段的包含字符

**FIELDS ESCAPED BY '字符'**: 设置转义字符, 默认为“\”

**LINES STARTING BY '字符串'**: 设置每行开头字符, 默认无字符

**LINES TERMINATED BY '字符串'**: 设置每行结束字符, 默认为“\N”, windows系统下, 换行符结束符为“\r\n”

例: `select * from student into outfile 'd:/tt.txt' fields terminated by '\', ' optionally enclosed by '\" lines starting by '\>' lines terminated by '\r\n'`





## 6.2.4 表的导入导出

### √ LOAD DATA INFILE 导入文本文件基本语法

**Load data [local] infile ‘文件路径’ into table 表名 [OPTION]**

**OPTION**选项:

**FIELDS TERMINATED BY ‘字符串’**: 设置字段的分隔符, 默认为“\t”

**FIELDS ENCLOSED BY ‘字符’**: 设置字段值包含在什么字符之中, 默认无字符

**FIELDS OPTIONALLY ENCLOSED BY ‘字符’**: 设置字符串类型字段的包含字符

**FIELDS ESCAPED BY ‘字符’**: 设置转义字符, 默认为“\”

**LINES STARTING BY ‘字符串’**: 设置每行开头字符, 默认无字符

**LINES TERMINATED BY ‘字符串’**: 设置每行结束字符, 默认为“\N”, windows系统下, 换行符结束符为“\r\n”

**IGNORE n LINES**: 忽略文件前n行

例: `load data local infile 'd:/tt.txt' into table student fields terminated by '\', ' optionally enclosed by '\" lines starting by '>' lines terminated by '\r\n'`



## 6.2.4 表的导入导出的字符集问题

√MYSQL中，字符集影响顺序如下：

- \* Character\_set\_server  $\hat{a}$
- \* Character\_set\_database  $\hat{a}$

```
mysql> show variables like 'character%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | gbk |
| character_set_connection | gbk |
| character_set_database | gbk |
| character_set_filesystem | binary |
| character_set_results | gbk |
| character_set_server | gbk |
| character_set_system | utf8 |
| character_sets_dir | C:\Program Files\MySQL\MySQL Server 5.5\share\charsets\ |
+-----+-----+
```

√修改字符集命令

- \* Set Character\_set\_database=gbk;





## 本章内容小结

### √ 6.1 MYSQL日志

- \* 6.1.1 错误日志
- \* 6.1.2 二进制日志
- \* 6.1.3 查询日志
- \* 6.1.4 慢日志

### √ 6.2 MYSQL备份与恢复

- \* 6.2.1 备份/恢复策略
- \* 6.2.2 逻辑备份和恢复
- \* 6.2.3 物理备份和恢复
- \* 6.2.4 表的导入导出

