

MySQL 数据库-操作基础

MySQL 官网: <https://www.mysql.com/>

1. MySQL 常见版本

- MySQL Community Server 社区版本, 开源免费, 但不提供官方技术支持。
- MySQL Enterprise Edition 企业版本, 需付费, 可以试用 30 天。
- MySQL Cluster 集群版, 开源免费。可将几个 MySQL Server 封装成一个 Server。
- MySQL Cluster CGE 高级集群版, 需付费

2. MySQL 安装部署

MySQL: MySQL 客户端程序

MySQL-Server: MySQL 服务器端程序

源代编译安装:

编译工具: configure、cmake、make

数据库常用的配置选项

-DCMAKE_INSTALL_PREFIX=/PREFIX	----指定安装路径 (默认的就是/usr/local/mysql)
-DMYSQL_DATADIR=/data/mysql	----mysql 的数据文件路径
-DSYSCONFDIR=/etc	----配置文件路径
-DWITH_INNOBASE_STORAGE_ENGINE=1	----使用 INNOBASE 存储引擎
-DWITH_READLINE=1	----支持批量导入 mysql 数据
-DWITH_SSL=system	----mysql 支持 ssl
-DWITH_ZLIB=system	----支持压缩存储
-DMYSQL_TCP_PORT=3306	----默认端口 3306
-DENABLED_LOCAL_INFILE=1	----启用加载本地数据
-DMYSQL_USER=mysql	----指定 mysql 运行用户
-DMYSQL_UNIX_ADDR=/tmp/mysql.sock	----默认套接字文件路径
-DEXTRA_CHARSETS=all	----是否支持额外的字符集
-DDEFAULT_CHARSET=utf8	----默认编码机制
-DWITH_DEBUG=0	----DEBUG 功能设置

常见资料:

服务: mysqld

端口: 3306

主配置文件: /etc/my.cnf

初始化脚本: mysql_install_db

启动命令: mysqld_safe

数据目录: /var/lib/mysql

套接字文件: /var/lib/mysql/mysql.sock

#当意外关闭数据库时，再开启时假如开启不了，找到这个，删除再启动

进程文件: /var/run/mysqld/mysqld.pid

MySQL 登录及退出命令:

设置密码: mysqladmin -uroot password '123456'

登录: mysql -u 用户名 -p 密码 -P 端口 -S 套接字文件

-p 用户密码

-h 登陆位置 (主机名或 ip 地址)

-P 端口号 (3306 改了就不是了)

-S 套接字文件 (/var/lib/mysql/mysql.sock)

退出命令: exit 或 ctrl+d

3. MySQL 管理命令

1. 创建登录用户

```
mysql>create user zhangsan@ '%' identified by '123456' ;
```

?:指任意的远程终端

2. 测试用户登录

```
# yum -y install mysql
```

```
# mysql -uzhangsan -p123456 -h 192.168.88.10
```

3. 用户为自己更改密码

```
mysql>set password=password ( '123456' );
```

4. root 用户为其他用户找回密码

```
mysql>set password for atguigu@ '%' =password( '123123' );
```

5. root 找回自己的密码并修改

关闭数据库, 修改主配置文件 (/etc/my.cnf) 添加: skip-grant-tables

```
# vim /etc/my.cnf
```

```
skip-grant-tables
```

启动数据库, 空密码登录并修改密码

```
update mysql.user set password=password( '新密码' ) where user=' root' ;
```

删除 skip-grant-tables, 重启数据库验证新密码

6. 创建查询数据库

```
mysql>create database web;
```

```
mysql>show databases;
```

7. 创建数据表

```
Mysql>use web;
```

#选择要使用的数据库

```
Mysql>create table a1 (id int ,name char(30));
```

#创建 a1 表，并添加 id 和 name 字段以及类型

```
Mysql>describe a1;
```

#查看表结构（字段）

复杂一点的

```
Mysql>create table a2 (
```

```
->id int unsigned not null auto_increment, #字段要求为正数、且自增长、主键
```

```
->name char(30) not null default ' ', #字符型长度 30 字节，默认值为空格
```

```
->age int not null default 0, #字段默认值为 0
```

```
->primary key (id)); #设置 id 为主键
```

```
Mysql> describe a2;
```

8. 插入数据

```
Mysql>insert into a2 (id,name,age) values (1, 'zhangsan' ,21); #指明插入字段和数据
```

```
Mysql>select * from a2;
```

```
Mysql>insert into a2 values (2, 'lisi' ,20); #按顺序插入指定字段
```

```
Mysql>insert into a2 values (3, 'wangwu' ); #未声明年龄
```

```
Mysql>insert into a2 values (4, 'zhao' ,19), (5, 'sun' ,25); #插入多条数据
```

9. 将表 a2 的数据复制到表 a1

```
Mysql>select * from a1;
```

```
Mysql>insert into a1 (id,name) select id,name from a2;
```

#查询 a2 值，并写入到 a1

```
Mysql>select * from a1;
```

10. 删除数据库

```
Mysql>drop database abc;
```

```
Mysql>show databases;
```

11. 删除数据表

```
Mysql>drop table a1;
```

```
Mysql>show table;
```

12. 删除表里的数据记录

```
Mysql>delete from a2 where id=5; #删除 id=5 的记录
```

```
Mysql>delete from a2 where between 23 and 25; #删除年龄在 23-25 之间的
```

注：库和表的删除用 drop，记录删除用 delete

13. 修改表中的数据

```
Mysql>update a2 set age=21 where id=3;
```

14. 修改数据表的名称

```
Mysql>alter table a2 rename a1;
```

15. 修改数据表的字段类型

```
Mysql>describe a1;  
Mysql>alter table a1 modify name char(50);  
Mysql>describe a1;
```

16. 修改数据表的字段类型详情

```
Mysql>describe a1;  
Mysql>alter table a1 change name username char(50) not null default '' ;  
Mysql>describe a1;
```

17. 添加字段

```
Mysql>describe a1;  
Mysql>alter table a1 add time datetime;  
Mysql>describe a1;  
#添加位置默认在末尾  
Mysql>alter table a1 add birthday year first; #添加字段到第一列  
Mysql>alter table a1 add sex nchar(1) after id; #添加到指定字段后
```

18. 删除字段

```
Mysql>alter table a1 drop birthday;
```

19. Mysql 用户授权

授予用户全部权限

```
Mysql>select user from mysql.user;  
Mysql>grant all on aa.a1 to atguigu@ '%' ; #给已存在用户授权  
Mysql>grant all on aa.a1 to abc@ '%' identified by '123456' ; #创建用户并授权
```

取消 abc 用户的删除库、表、表中数据的权限

```
Mysql>revoke drop,delete on aa.a1 from abc@ '%' ; #取消删除权限（登录 abc 测试）  
Mysql>show grants for abc@ '%' ; #查看指定用户的授权  
Mysql>show grants for atguigu@ '%' ;
```

4. 备份和还原

mysqldump 备份:

备份:

```
mysqldump -u 用户名 -p 数据库名 > /备份路径/备份文件名（备份整个数据库）
```

`mysqldump -u 用户名 -p 数据库名 表名 > /备份路径/备份文件名 (备份数据表)`

备份多个库: `--databases 库 1, 库 2`

备份所有库: `--all-databases`

备份多个表: 库名 表 1 表 2

还原: `mysql 数据库 < 备份文件`

注意: 还原时, 若导入的是某表, 请指定导入到哪一个库中

mysqlhotcopy 备份:

备份: `mysqlhotcopy --flushlog -u=' 用户' -p=' 密码' --regex=正则 备份目录`

还原: `cp -a 备份目录 数据目录 (/var/lib/mysql)`

mysqldump 和 mysqlhotcopy 示例:

Mysql 备份和还原

把数据库 aa 备份到/root 目录下

```
# mysqldump -uroot -p aa > ~/aa.sql
```

模拟数据库 aa 丢失 (删除数据库 aa)

```
# Mysql>drop database aa;
```

通过 aa.sql 文件还原 (指定导入到哪个库中)

```
# mysql -uroot -p test < aa.sql
```

备份多个数据库 (`--databases`)

```
# mysqldump -uroot -p --databases aa test > abc.sql
```

还原 (先模拟丢失)

```
# mysql -uroot -p < abc.sql
```

备份有规则的数据库

```
Mysql>create database a1; #连续创建三个 a 开头的数据库
```

```
# mysqlhotcopy --flushlog -u='root' -p='456' --regex=a
```

还原 (先模拟丢失)

```
Mysql>drop database a1; #顺序删除 a 开头的数据库
```

```
# cp -a /mnt/* /var/lib/mysql/ #复制产生的文件到数据库目录下
```

```
#登录数据库查看即可
```

mysql-binlog 日志备份:

二进制日志 (`log-bin` 日志): 所有对数据库状态更改的操作 (`create`、`drop`、`update` 等)

修改 `my.cnf` 配置文件开启 binlog 日志记录功能

```
# vim /etc/my.cnf
```

```
log-bin=mysql-bin #启动二进制日志
```

按时间还原:

```
--start-datetime
```

```
--stop-datetime
```

格式: `mysqlbinlog --start-datetime 'YY-MM-DD HH:MM:SS' --stop-datetime 'YY-MM-DD`

HH:MM:SS' 二进制日志 | `mysql -uroot -p`

按文件大小还原:

`--start-position`

`--stop-position`

mysql-binlog 日志备份示例:

开启二进制日志

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted s
symbolic-links=0
general-log=0N
log=0N
log-slow-queries=/var/log/mysql-slow.log
log-bin=mysql-bin 在主配置文件中添加
```

查看二进制日志文件

```
[root@localhost ~]# cd /var/lib/mysql/
[root@localhost mysql]# ls
aa cc ib_logfile0 mysql mysql-bin.index ON
bb ibdata1 ib_logfile1 mysql-bin.000001 mysql.sock test
[root@localhost mysql]#
[root@localhost mysql]#
[root@localhost mysql]#
```

在数据目录下会产生二进制日志文件

按时间还原:

如果数据库中的 bb 库被删, 需要还原

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| aa |
| cc |
| mysql |
| test |
+-----+
5 rows in set (0.00 sec)
```

bb库被删除

查看二进制日志内容

```
/*!*/;
# at 185
#170225 0:35:40 server id 1 end_log_pos 264 Query thread_id=2 e
xec_time=0 error_code=0
SET TIMESTAMP=1487954140/*!*/;
create database bb
/*!*/;
# at 264
#170225 0:35:53 server id 1 end_log_pos 341 Query thread_id=2 e
xec_time=0 error_code=0
SET TIMESTAMP=1487954153/*!*/;
drop database bb
/*!*/;
```

mysqlbinlog mysql-bin.000001 查看二进制日志的内容如下一部分:

找到丢失的地方, bb库被删, 那么找到创建它的时间和删除时的时间

还原并查看

```
mysqlbinlog --start-datetime='2018-09-11 14:24:00' --stop-datetime='2018-09-11 14:28:00'  
mysql-bin.000006 | mysql -uroot -p123123
```

注：所选时间段一定要完整包含所有动作（可以在原来基础上稍微增加点时间）

按文件大小还原：还原到 bb 库被删除的数据状态

1. 查看 bb 库被删除前后的文件大小

```
# at 264  
#170225 0:35:53 server id 1  end_log_pos 341  Query  thread_id=2  e  
xec_time=0      error_code=0  
SET TIMESTAMP=1487954153/*!*/;  
drop database bb  
/*!*/;  
# at 341  
#170225 0:36:16 server id 1  end_log_pos 420  Query  thread_id=2  e  
xec_time=0      error_code=0
```

还原并查看