# 数据备份与还原

## 数据备份

在MySQL中使用mysqldump命令实现数据的备份。

1. 备份单个数据库（或单个或多个表），其语法格式为：

**mysqldump -u 用户名 -p dbname[ tbname1 tbname2 …]>filename.sql**

dbname表示需要备份的数据库的名称；

tbname表示数据库中表名称，可以指定一个或多个数据表，如不指定，则备份整 个数据库；

filename表示备份文件的名称，文件名之前可以加上本机的绝对路径，如果不加绝对路径，则生成的备份文件将保存在本机用户目录下。

1. 备份多个数据库，其语法格式为：

**mysqldump -u 用户名 -p --database dbname1 dbname2 …>filename.sql**

--database参数后面可以指定一个或多个数据库名称。

1. 备份所有数据库，其语法格式为：

**mysqldump -u 用户名 -p --all-database>filename.sql**

备份数据库命令是直接在命令行上完成的，不需要登录到客户端。注意命令后面不能使 用分号结束。

## 数据还原

当数据库中的数据遭到破坏时，可以通过备份好的数据文件进行还原。还原数据有两种 方式：

1. 使用mysql命令完成。其语法格式为：

**mysql -u 用户名 -p 数据库名<filename.sql**

在命令行上完成，不需要登录客户端，但指定的数据库必须存在。（注意：filename.sql可以包含绝对路径）

1. 使用source命令完成。其语法格式为：

**source filename.sql ;**

使用source命令还原数据需要登录到客户端完成，且必须先使用use命令选择一个数据库再进行还原操作。（注意：filename.sql可以包含绝对路径）

# 用户管理

在MySQL中的用户分为root用户和普通用户，root用户为超级管理员，拥有一切权限。 而普通用户只拥有被赋予的某些权限。

## 创建用户

创建普通用户的方式有三种：

1. 使用GRANT语句创建。其语法格式为：

**GRANT 权限列表 ON database.table TO ‘用户名’@’主机号’**

**IDENTIFIED BY ‘密码’;**

使用GRAND语句会创建用户自动加载权限表，而且安全、准确、错误少，是最常用 的方法。

1. 使用CREATE USER语句创建。其语法格式为：

**CREATE USER ‘用户名’@’主机号’ IDENTIFIED BY ‘密码’ ;**

使用CREATE USER语句创建用户时服务器会自动修改相应的授权表，但该语句创建 的新用户是没有任何权限的。

1. \*使用INSERT语句创建。其语法格式为：

**INSERT INTO mysql.user(Host,User,Password,ssl\_cipher,x509\_issuer,x509\_subject)**

**VALUES(‘主机号’ , ’用户名’ , PASSWORD(‘密码’) , ’ ’ , ’ ’ , ’ ’) ;**

由于INSERT语句是针对user表直接操作的，所有INSERT语句没有自动刷新权限的 功能，需要使用**FLUSH PRIVILEGES;**语句手动刷新当前权限表或重新启动MySQL服 务器。此种方式是最麻烦的。

## 查看用户

MySQL中的用户信息存放在mysql数据库中的user数据表中的，如果需要查看MySQL有多少个用户，则直接查询user表即可：

**SELECT host,name,password FROM user ;**

当然，user表远不止有host、name、password这三个字段；

## 删除用户

如果发现某些用户时是没有必要的，就可以将其删除。删除用户有两种方式。

1. 使用DROP USER语句删除用户。其语法格式为：

**DROP USER ‘用户名’@’主机号’ ;**

1. 使用DELETE语句删除用户。其语法格式为：

**DELETE FROM mysql.user WHERE host=’主机号’ AND user=’用户名’ ;**

使用DELETE语句删除用户必须拥有对mysql.user表的DELETE权限。

由于DELETE语句是直接对uers表进行操作，执行命令后需要使用FLUSH PRIVILEGES; 语句重新加载用户权限或重新启动数据库服务器。

## 修改root用户的密码

只有root用户才能修改自己的密码。修改root用户密码有两种方式。

1. 使用mysqladmin命令修改root用户密码。其语法格式为：

**mysqladmin -u username [-h hostname] -p password newpassword**

username为要修改的用户名；

-h参数表示对应的主机号，如不指定，则默认为localhost；

password表示密码关键字；

newpassword表示新密码。

但使用此种方式修改密码需要在C:\Administrator目录下进行修改。无需登录到客 户端。

1. 使用UPDATE语句修改root用户密码。由于所有的用户信息都存放在mysql.user表 中，因此只要root用户登录到MySQL客户端，使用UPDATE语句就可以直接修改自 己的密码。其语法格式为：

**UPDATE mysql.user SET password=PASSWORD(‘newpassword’)**

**WHERE user=’root’ AND host=’localhost’ ;**

但由于UPDATE语句是直接对表进行操作的，不能刷新权限表，所以修改完密码后 应使用FLUSH PRIVILEGES语句重新加载用户权限，否则只有重新启动服务器修改的 密码才会生效。

1. 使用SET语句修改密码。其语法格式为：

**SET PASSWORD = PASSWORD(‘ newpassword ’) ;**

PASSWORD( )用于给新密码加密，并且使用SET语句会自动重新加载用户权限。

## root用户修改普通用户的密码

root用户除了修改自己的密码，还可以修改普通用户的密码。root用户修改普通用 户 密码有两种方式。

1. 使用GRANT用户修改普通用户的密码。GRANT语句不仅可以创建用户为用户授权， 还可以修改用户的密码。其语法格式为：

**GRANT USAGE ON \*.\* TO ‘username’@’localhost’ IDENTIFIED BY ‘newpassword’ ;**

1. 使用UPDATE语句修改普通用户的密码。其语法格式为：

**UPDATE mysq.user SET password=PASSWORD(‘ newpassword ‘)**

**WHERE user=’username’ AND host=’hostname’ ;**

但需要使用FLUSH PRIVILEGES语句重新加载权限表或者重启服务器后才生效。

1. 使用SET语句修改普通用户的密码。其语法格式为：

**SET PASSWORD FOR ‘username’@’hostname’=PASSWORD(‘ newpassword ’) ;**

## 普通用户修改自己的密码

普通用户登录到客户端后也可以通过SET语句来修改自己的密码。其语法格式为：

**SET PASSWORD=PASSWORD(‘newpassword’) ;**

## 解决root用户密码丢失

root用户是超级管理员，如果其密码一旦丢失，就会造成很大的麻烦。为次，MySQL 提供了对应的处理机制，可以通过特殊方法登录到MySQL服务器，然后重新为root用 户设置密码。操作步骤如下：

1. 关闭MySQL服务器
2. 在命令行（管理员身份打开的）使用mysqld --skip-grant-tables启动MySQL服务器
3. 重新开启一个命令行，输入‘mysql -u root’登录到服务器
4. 使用UPDATE语句设置密码（见上）
5. 使用**FLUSH PRIVILEGES ;**语句重新加载权限表
6. 退出MySQL客户端，并将刚刚启动MySQL服务器的命令行关闭，然后以正常方式启动MySQL服务器，打开一个命令行重新登录MySQL。

**权限管理：**

在MySQL中，为了保证用户的安全性，数据库管理员需要为每一个用户赋予不同的权 限，以满足不同用户的需求。

MySQL中权限信息被存储在mysql数据库的user、db、host、tables\_priv、column\_pri 和procs\_priv表中，当启动MySQL时会自动加载这些权限信息并将它们读到内存中。 常见的权限有：

1. CREATE和DROP权限，可以创建数据库、表、索引，或者删除已有的数据库、表、 索引。
2. INSERT、DELETE、UPDATE、SELECT权限，可以对数据库中的表进行增删查改操作。
3. INDEX权限，可以创建或删除索引，适用于所有的表。
4. ALTER权限，可以用于修改表的结构或重命名表。
5. FILE权限，被赋予该权限的用户能读写MySQL服务器上的任何文件。

# 用户权限管理

## 授予权限

使用GRANT语句在创建用户时为用户授权。其语法格式为：  
 **GRANT privileges [(columns)],... ON database.table**

**To ‘username’@’hostname’ IDENTIFIED BY ‘password’**

**[WITH with\_option1,...];**

1. privileges表示权限列表，columns参数表示权限作用域某一列，如省略，则权限作 用于整个表。
2. database.table表示用户权限范围作用于哪个数据库中的数据表。如\*.\*表示作用于整 个数据库系统，如a.\*表示作用于a数据库，如a.student表示作用于a数据库的student 表。
3. username表示用户名，hostname表示主机号，IDENTIFIED BY表示为用户设置密码。 password表示设置的密码。
4. WITH关键字为可选参数，WITH关键字后面可以带多个参数，这个参数有5个取值。 为：

·GRANT OPTION：将自己的权限授予给其他用户。

·MAX\_QUERIES\_PER\_HOUR count：设置每个用户最多可以执行多少次(count) 查询。

·MAX\_UPDATES\_PER\_HOUR count：设置每小时最多可以执行多少次更新。

·MAX\_COUNNECTIONS\_PER\_HOUR count：设置每小时最大的连接数量。

·MAX\_USER\_CONNECTIONS：设置每个用户最多可以同时建立连接的数量。

## 查询权限

可以使用SELECT语句查询mysql库中的user表可以获取用户的权限信息，但也存在一种专门用于查询指定用户权限信息的语法。其语法格式为：

**SHOW GRANT FOR ‘username’@’localhost’ \G;**

username表示用户名，localhost表示主机号。

使用这种方式查看用户权限是通过查看GRANT语句创建用户的语句来得知用户权限的。

## 收回权限

为了保证数据的安全性，需要将用户不必要的权限收回。在MySQL中使用REVOKE语 句收回用户的权限。其语法格式为：

**REVOKE privileges[columns],... ON database.table**

**FROM ‘username’@’hostname’ ;**

1. privileges表示要收回的权限列表。
2. database.table表示要回收作用于哪个数据库及表的权限。
3. username表示用户名，hostname表示主机号。