# 概述

DCL语句主要是DBA用来管理系统中的对象权限时使用，一般的开发人员很少使用。



# 权限表

mysql数据库中的3个权限表：user 、db、 host

权限表的存取过程是：

1、先从user表中的host、 user、 password这3个字段中判断连接的IP、用户名、密码是否存在表中，存在则通过身份验证；

2、通过权限验证，进行权限分配时，按照user、db、tables\_pri、columns\_priv的顺序进行分配。即先检查全局权限表user，如果user中对应的权限为Y，则此用户对所有数据库的权限都为Y，将不再检查db，tables\_priv，columns\_priv；如果为N，则到db表中检查此用户对应的具体数据库，并得到db中为Y的权限；如果db中为N，则检查tables\_priv中此数据库对应的具体表，取得表中的权限Y，以此类推。

# 权限

以下操作都是以root身份登陆进行grant授权，以p1@localhost身份登陆执行各种命令。

## usage

连接（登陆）权限，建立一个用户，就会自动授予其usage权限（默认授予）。

mysql> grant usage on \*.\* to ‘p1′@’localhost’ identified by ‘123′;

该权限只能用于数据库登陆，不能执行任何操作；且usage权限不能被回收，也即REVOKE用户并不能删除用户。

## select

必须有select的权限，才可以使用select table

mysql> grant select on pyt.\* to ‘p1′@’localhost’;

mysql> select \* from shop;

## create

必须有create的权限，才可以使用create table

mysql> grant create on pyt.\* to ‘p1′@’localhost’;

## create routine

必须具有create routine的权限，才可以使用{create |alter|drop} {procedure|function}

mysql> grant create routine on pyt.\* to ‘p1′@’localhost’;

当授予create routine时，自动授予EXECUTE, ALTER ROUTINE权限给它的创建者。

## create temporary tables

必须有create temporary tables的权限（注意这里是tables，不是table），才可以使用create temporary tables。

mysql> grant create temporary tables on pyt.\* to ‘p1′@’localhost’;

[mysql@mydev ~]$ mysql -h localhost -u p1 -p pyt

mysql> create temporary table tt1(id int);

## create view

必须有create view的权限，才可以使用create view

mysql> grant create view on pyt.\* to ‘p1′@’localhost’;

mysql> create view v\_shop as select price from shop;

## create user

要使用CREATE USER，必须拥有mysql数据库的全局CREATE USER权限，或拥有INSERT权限。

mysql> grant create user on \*.\* to ‘p1′@’localhost’;

或：mysql> grant insert on \*.\* to p1@localhost;

## insert

必须有insert的权限，才可以使用insert into ….. values….

## alter

必须有alter的权限，才可以使用alter table

alter table shop modify dealer char(15);

## alter routine

必须具有alter routine的权限，才可以使用{alter |drop} {procedure|function}

mysql>grant alter routine on pyt.\* to ‘p1′@’ localhost ‘;

mysql> drop procedure pro\_shop;

Query OK, 0 rows affected (0.00 sec)

mysql> revoke alter routine on pyt.\* from ‘p1′@’localhost’;

[mysql@mydev ~]$ mysql -h localhost -u p1 -p pyt

mysql> drop procedure pro\_shop;

ERROR 1370 (42000): alter routine command denied to user ‘p1′@’localhost’ for routine ‘pyt.pro\_shop’

## update

必须有update的权限，才可以使用update table

mysql> update shop set price=3.5 where article=0001 and dealer=’A';

## delete

必须有delete的权限，才可以使用delete from ….where….(删除表中的记录)

## drop

必须有drop的权限，才可以使用drop database db\_name; drop table tab\_name;

drop view vi\_name; drop index in\_name;

## show database

通过show database只能看到你拥有的某些权限的数据库，除非你拥有全局SHOW DATABASES权限。

对于p1@localhost用户来说，没有对mysql数据库的权限，所以以此身份登陆查询时，无法看到mysql数据库：

mysql> show databases;

+——————–+

| Database |

+——————–+

| information\_schema|

| pyt |

| test |

+——————–+

## show view

必须拥有show view权限，才能执行show create view。

mysql> grant show view on pyt.\* to p1@localhost;

mysql> show create view v\_shop;

## index

必须拥有index权限，才能执行[create |drop] index

mysql> grant index on pyt.\* to p1@localhost;

mysql> create index ix\_shop on shop(article);

mysql> drop index ix\_shop on shop;

## excute

执行存在的Functions,Procedures

mysql> call pro\_shop1(0001,@a)；

+———+

| article |

+———+

| 0001 |

| 0001 |

+———+

mysql> select @a;

+——+

| @a |

+——+

| 2 |

+——+

## lock tables

必须拥有lock tables权限，才可以使用lock tables

mysql> grant lock tables on pyt.\* to p1@localhost;

mysql> lock tables a1 read;

mysql> unlock tables;

## references

有了REFERENCES权限，用户就可以将其它表的一个字段作为某一个表的外键约束。

## reload

必须拥有reload权限，才可以执行flush [tables | logs | privileges]

mysql> grant reload on pyt.\* to p1@localhost;

ERROR 1221 (HY000): Incorrect usage of DB GRANT and GLOBAL PRIVILEGES

mysql> grant reload on \*.\* to ‘p1′@’localhost’;

Query OK, 0 rows affected (0.00 sec)

mysql> flush tables;

## replication client

拥有此权限可以查询master server、slave server状态。

mysql> show master status;

ERROR 1227 (42000): Access denied; you need the SUPER,REPLICATION CLIENT privilege for this operation

mysql> grant Replication client on \*.\* to p1@localhost;

或：mysql> grant super on \*.\* to p1@localhost;

mysql> show master status;

+——————+———-+————–+——————+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+——————+———-+————–+——————+

| mysql-bin.000006 | 2111 | | |

+——————+———-+————–+——————+

mysql> show slave status;

## replication slave

拥有此权限可以查看从服务器，从主服务器读取二进制日志。

mysql> show slave hosts;

ERROR 1227 (42000): Access denied; you need the REPLICATION SLAVE privilege for this operation

mysql> show binlog events;

ERROR 1227 (42000): Access denied; you need the REPLICATION SLAVE privilege for this operation

mysql> grant replication slave on \*.\* to p1@localhost;

mysql> show slave hosts;

Empty set (0.00 sec)

mysql>show binlog events;

+—————+——-+—————-+———–+————-+————–+

| Log\_name | Pos | Event\_type | Server\_id| End\_log\_pos|Info |

+—————+——-+————–+———–+————-+—————+

| mysql-bin.000005 | 4 | Format\_desc | 1 | 98 | Server ver: 5.0.77-log, Binlog ver: 4 | |mysql-bin.000005|98|Query|1|197|use `mysql`; create table a1(i int)engine=myisam|

……………………………………

## Shutdown

关闭MySQL：

[mysql@mydev ~]$ mysqladmin shutdown

重新连接：

[mysql@mydev ~]$ mysql

ERROR 2002 (HY000): Can’t connect to local MySQL server through socket ‘/tmp/mysql.sock’ (2)

[mysql@mydev ~]$ cd /u01/mysql/bin

[mysql@mydev bin]$ ./mysqld\_safe &

[mysql@mydev bin]$ mysql

## grant option

拥有grant option，就可以将自己拥有的权限授予其他用户（仅限于自己已经拥有的权限）

mysql> grant Grant option on pyt.\* to p1@localhost;

mysql> grant select on pyt.\* to p2@localhost;

## file

拥有file权限才可以执行 select ..into outfile和load data infile…操作，但是不要把file, process, super权限授予管理员以外的账号，这样存在严重的安全隐患。

mysql> grant file on \*.\* to p1@localhost;

mysql> load data infile ‘/home/mysql/pet.txt’ into table pet;

为用户赋予file权限时不必指明该权限专门赋予哪一个schema，若指明schema，则会报错。

mysql> grant file on test\_db.\* to "test\_user"@"%";

ERROR 1221 (HY000): Incorrect usage of DB GRANT and GLOBAL PRIVILEGES

若在赋予file权限时不对schema作限制，则可以赋权成功。

mysql> grant file on \*.\* to "test\_user"@"%";

Query OK, 0 rows affected (0.00 sec)

## super

这个权限允许用户终止任何查询；修改全局变量的SET语句；使用CHANGE MASTER，PURGE MASTER LOGS。

mysql> grant super on \*.\* to p1@localhost;

mysql> purge master logs before ‘mysql-bin.000006′;

## process

通过这个权限，用户可以执行SHOW PROCESSLIST和KILL命令。默认情况下，每个用户都可以执行SHOW PROCESSLIST命令，但是只能查询本用户的进程。

mysql> show processlist;

+—-+——+———–+——+———+——+——-+——————+

| Id | User | Host | db | Command | Time | State | Info |

+—-+——+———–+——+———+——+——-+——————+

| 12 | p1 | localhost | pyt | Query | 0 | NULL | show processlist |

+—-+——+———–+——+———+——+——-+——————+

另外，管理权限（如 super， process， file等）不能够指定某个数据库，on后面必须跟\*.\*

mysql> grant super on pyt.\* to p1@localhost;

ERROR 1221 (HY000): Incorrect usage of DB GRANT and GLOBAL PRIVILEGES

mysql> grant super on \*.\* to p1@localhost;

Query OK, 0 rows affected (0.01 sec)

# 操作

## create user

## grant

## drop user

删除用户及目录：userdel -r user1

注：这是Linux的命令，不是mysql的命令。

## plush privileges

# 分布式数据库实践

## 计算节点与数据节点权限不统一

在分布式数据库中，由于计算节点没有真正的用户权限，生效的是DB的权限表，所以会出现这样的一种现象：

1. 计算节点下发select into outfile
2. 数据节点不设置file权限

但是却可以正常执行语句，原因在于计算节点会将select into outfile拆分成select下发到各个DB，然后在计算节点做into outfile的数据汇聚，而DB是有select权限的，所以可以正常执行。

解决方法：计算节点引入权限表的概念，并且DB数据节点检查发过来sql的权限类型。

## 并发创建用户产生元数据锁

在分布式数据库中，计算节点不支持并发创建用户，比如同时复制以下语句执行会报错：

create user user1 identified by ‘12345’;

grant all on \*.\* to user1;

create user user2 identified by ‘12345’;

grant all on \*.\* to user2;

报错信息类似：other use own lock

之所以这样是因为，防止并发执行用户创建DDL造成这个分片之间的密码、权限不一致。

解决方法：

1. 去元数据的数据库查询锁状态
2. 然后su元数据用户下执行解锁