授予权限

创建新的用户帐户后，用户没有任何权限。需要使用 GRANT 语句向用户帐户授予权限。

```
GRANT privilege,[privilege],.. ON privilege_level
TO user [IDENTIFIED BY password]
 [WITH [GRANT_OPTION | resource_option]];
```

语法说明：

1.在 GRANT 关键字之后 privilege 指定一个或多个特权。如果要授予用户多个权限，则每个权限都将以逗号分隔。

2.指定确定特权应用级别的 privilege_level。MySQL 支持全局(*.*)，数据库(database.*)，表(database.table)和列级别。 如果您使用列权限级别，则必须在每个权限之后使用逗号分隔列的列表。

3. TO user 为放置要授予权限的用户。如果用户已经存在，则 GRANT 语句修改其特权。如不存在，则 GRANT 语句将创建一个新用户。可选的条件 IDENTIFIED BY 允许为用户设置新密码。

4.可选的 WITH GRANT OPTION 子句允许此用户授予其他用户或从其他用户删除被拥有的权限。

5.可以使用 WITH resource_option 子句来分配 MySQL 数据库服务器的资源，例如，设置用户每小时可以使用多少个连接或语句。这在 MySQL 共享托管等共享环境中非常有用。

6.要使用 GRANT 语句，用户必须具有 GRANT OPTION 权限和授予其它用户的权限。 如果启用了 read_only 系统变量，则需要具有 SUPER 权限才能执行 GRANT 语句授权。

MySQL 可授予的权限列表：

权 限	作用范围	作 用
all	服务器	所有权限
select	表、列	选择行
insert	表、列	插入行
update	表、列	更新行
delete	表	删除行
create	数据库、表、索引	创建
drop	数据库、表、视图	删除
reload	服务器	允许使用 flush 语句
shutdown	服务器	关闭服务
process	服务器	查看线程信息
file	服务器	文件操作
grant option	数据库、表、存储过程	授权
references	数据库、表	外键约束的父表
index	表	创建/删除索引
alter	表	修改表结构
show databases	服务器	查看数据库名称
super	服务器	超级权限
create temporary tables	表	创建临时表
lock tables	数据库	锁表
execute	存储过程	执行
replication client	服务器	允许查看主/从/二进制日志状态

replication slave	服务器	主从复制
create view	视图	创建视图
show view	视图	查看视图
create routine	存储过程	创建存储过程
alter routine	存储过程	修改/删除存储过程
create user	服务器	创建用户
event	数据库	创建/更改/删除/查看事件
trigger	表	触发器
create tablespace	服务器	创建/更改/删除表空间/日志文件
proxy	服务器	代理成为其它用户
usage	服务器	没有权限

GRANT 语句功能强大，授与的权限可以分为多个层级：

1.全局层级

全局权限适用于 1 个给定服务器中的所有数据库。这些权限存储在 mysql.user 表中。GRANT ALL ON .和 REVOKE ALL ON .只授与和撤消全局权限。

【例 9-15】授予 user1 查询 MySQL 中所有数据库中表的权限。在终端窗口输入命令：

```
GRANT SELECT on *.* to user1@localhost;
SHOW GRANTS FOR 'user1'@'localhost';
```

```
mysql> grant select on *.* to user1@localhost;
Query OK, 0 rows affected (0.10 sec)

mysql> SHOW GRANTS FOR 'user1'@'localhost';
+-------------------------------------------------------------+
| Grants for user1@localhost                                  |
+-------------------------------------------------------------+
| GRANT SELECT ON *.* TO 'user1'@'localhost'                  |
| GRANT SELECT ON `bookstore`.`book` TO 'user1'@'localhost'   |
+-------------------------------------------------------------+
2 rows in set (0.00 sec)
```

2.数据库层级

数据库权限适用于 1 个给定数据库中的所有目标。这些权限存储在 mysql.db 和 mysql.host 表中。GRANT ALL ON db_name.和 REVOKE ALL ON db_name.只授与和撤消数据库权限。

【例 9-16】授予 user1 在 lib 数据库中所有表新增数据的权限。在终端窗口输入命令：

```
GRANT INSERT on lib.* to user1@localhost;
SHOW GRANTS FOR 'user1'@'localhost';
```

```
mysql> GRANT INSERT on lib.* to user1@localhost;
Query OK, 0 rows affected (0.04 sec)

mysql> SHOW GRANTS FOR 'user1'@'localhost';
+-----------------------------------------------------------+
| Grants for user1@localhost                                |
+-----------------------------------------------------------+
| GRANT SELECT ON *.* TO 'user1'@'localhost'                |
| GRANT INSERT ON `lib`.* TO 'user1'@'localhost'            |
| GRANT SELECT ON `bookstore`.`book` TO 'user1'@'localhost' |
+-----------------------------------------------------------+
3 rows in set (0.00 sec)
```

3.表层级

表权限适用于 1 个给定表中的所有列。这些权限存储在 mysql.talbes_priv 表中。GRANT ALL ON db_name.tbl_name 和 REVOKE ALL ON db_name.tbl_name 只授与和撤消表权限。

【例 9-17】授予 user1 在 lib 数据库中 book 表修改数据的权限。在终端窗口输入命令：

```
GRANT UPDATE on lib.book to user1@localhost;
SHOW GRANTS FOR 'user1'@'localhost';
```

```
mysql> GRANT UPDATE on lib.book to user1@localhost;
Query OK, 0 rows affected (0.07 sec)

mysql> SHOW GRANTS FOR 'user1'@'localhost';
+-----------------------------------------------------------+
| Grants for user1@localhost                                |
+-----------------------------------------------------------+
| GRANT SELECT ON *.* TO 'user1'@'localhost'                |
| GRANT INSERT ON `lib`.* TO 'user1'@'localhost'            |
| GRANT SELECT ON `bookstore`.`book` TO 'user1'@'localhost' |
| GRANT UPDATE ON `lib`.`book` TO 'user1'@'localhost'       |
+-----------------------------------------------------------+
4 rows in set (0.00 sec)
```

4.列层级

列权限适用于 1 个给定表中的单 1 列。这些权限存储在 mysql.columns_priv 表中。当使用 REVOKE 时，必须指定与被授权列相同的列。

【例 9-18】授予 user1 在 lib 数据库 borrow 表中修改“borrowDate”和“returnDate”数据的权限。在终端窗口输入命令：

```
GRANT UPDATE(borrowDate,returnDate) on lib.borrow to user1@localhost;
SHOW GRANTS FOR 'user1'@'localhost';
```

```
mysql> GRANT UPDATE(borrowDate,returnDate) on lib.borrow to user1@localhost;
Query OK, 0 rows affected (0.05 sec)

mysql> SHOW GRANTS FOR 'user1'@'localhost';
+-----------------------------------------------------------------------------------+
| Grants for user1@localhost                                                        |
+-----------------------------------------------------------------------------------+
| GRANT SELECT ON *.* TO 'user1'@'localhost'                                        |
| GRANT INSERT ON `lib`.* TO 'user1'@'localhost'                                    |
| GRANT SELECT ON `bookstore`.`book` TO 'user1'@'localhost'                         |
| GRANT UPDATE ON `lib`.`book` TO 'user1'@'localhost'                               |
| GRANT UPDATE (returnDate, borrowDate) ON `lib`.`borrow` TO 'user1'@'localhost'    |
+-----------------------------------------------------------------------------------+
5 rows in set (0.00 sec)
```

验证权限需要以 user1 用户登录

【例 9-19】修改第 499 条借书记录的借书日期为“2020-01-01”，还书日期为“2020-04-01”；

Update borrow set borrowDate=’2020-01-01’,returnDate=’2020-04-01’ where borrowNo=499;

select * from borrow where borrowNo=499;

```
mysql> Update borrow set borrowDate='2020-01-01',returnDate='2020-04-01' where borrowNo=499;
Query OK, 1 row affected (0.11 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from borrow where borrowNo=499;
+----------+--------+----------+---------------------+---------------------+--------+---------+
| borrowNo | bookNo | readerNo | borrowDate          | returnDate          | re_new | re_turn |
+----------+--------+----------+---------------------+---------------------+--------+---------+
|      499 | b111   | r096     | 2020-01-01 00:00:00 | 2020-04-01 00:00:00 | 否     | 是      |
+----------+--------+----------+---------------------+---------------------+--------+---------+
1 row in set (0.00 sec)
```

【例 9-20】修改第 499 条借书记录的 bookNo 为“b110”，还书日期为“2020-04-01”；

Update borrow set bookNo=’b110’ where borrowNo=499;

```
mysql> Update borrow set bookNo='b110' where borrowNo=499;
ERROR 1143 (42000): UPDATE command denied to user 'user1'@'localhost' for column 'bookNo' in table 'borrow'
```

5.子程序层级

CREATE ROUTINE, ALTER ROUTINE, EXECUTE 和 GRANT 权限适用于已存储的子程序。这些权限可以被授与为全局层级和数据库层级。而且，除 CREATE ROUTINE 外，这些权限可以被授与为子程序层级，并存储在 mysql.procs_priv 表中。

【例 9-21】授予 user1 在 lib 数据库中执行存储过程 pr_add 的权限。在终端窗口输入命令：

grant execute on procedure lib.pr_add to 'user1'@'localhost';

6.权限的转移

GRANT 语句最后还有可选的 WITH GRANT OPTION 子句，它将允许此用户授予其他用户或从其他用户删除被拥有的权限。

【例 9-22】授予 user1 在 lib 数据库中所有表新增数据的权限，并允许其将该授权授予其他用户。在终端窗口输入命令：。

GRANT INSERT on lib.* to user1@localhost with grant option;

```
mysql> GRANT INSERT on lib.* to user1@localhost with grant option;
Query OK, 0 rows affected (0.00 sec)
```

然后以 user1 身份登录 MYSQL，将其拥有的新增数据的权限传递给 user2。

GRANT INSERT on lib.* to user2@localhost;

```
mysql> GRANT INSERT on lib.* to user2@localhost;
Query OK, 0 rows affected (0.00 sec)
```

回到 root 用户下查看 user1，user2 的操作权限。

SHOW GRANTS FOR 'user1'@'localhost';

SHOW GRANTS FOR 'user2'@'localhost';

```
mysql> use mysql;
Database changed
mysql> SHOW GRANTS FOR 'user1'@'localhost';
+-----------------------------------------------------------------------------------------+
| Grants for user1@localhost                                                              |
+-----------------------------------------------------------------------------------------+
| GRANT SELECT ON *.* TO 'user1'@'localhost'                                              |
| GRANT INSERT ON `lib`.* TO 'user1'@'localhost' WITH GRANT OPTION                        |
| GRANT SELECT ON `bookstore`.`book` TO 'user1'@'localhost'                               |
| GRANT UPDATE ON `lib`.`book` TO 'user1'@'localhost'                                     |
| GRANT UPDATE (returnDate, borrowDate) ON `lib`.`borrow` TO 'user1'@'localhost'          |
+-----------------------------------------------------------------------------------------+
5 rows in set (0.00 sec)

mysql> SHOW GRANTS FOR 'user2'@'localhost';
+-----------------------------------------------------------+
| Grants for user2@localhost                                |
+-----------------------------------------------------------+
| GRANT USAGE ON *.* TO 'user2'@'localhost'                 |
| GRANT INSERT ON `lib`.* TO 'user2'@'localhost'            |
| GRANT SELECT ON `bookstore`.`book` TO 'user2'@'localhost' |
+-----------------------------------------------------------+
3 rows in set (0.00 sec)
```

7.权限的限制

可选的 WITH resource_option 子句可以来分配 MySQL 数据库服务器的资源。

MAX_USER_CONNECTIONS：全局变量，一个用户可以在同一时间连接 MySQL 实例的数量，此参数无法对每个用户区别对待。

MAX_QUERIES_PER_HOUR：一个用户在一个小时内可以执行查询的次数（基本包含所有语句）。

MAX_UPDATES_PER_HOUR：一个用户在一个小时内可以执行修改的次数（仅包含修改数据库或表的语句）。

MAX_CONNECTIONS_PER_HOUR：一个用户在一个小时内可以连接 MySQL 的时间。

后面三个参数后接次数为 0，则表示不起限制作用。

【例 9-23】授予 user1 在 lib 数据库中每小时只能处理 10 条 insert 语句的权限。在终端窗口输入命令：

grant insert on lib.* to 'user1'@'localhost' with MAX_QUERIES_PER_HOUR 10;

```
mysql> grant insert on lib.* to 'user1'@'localhost' with MAX_QUERIES_PER_HOUR 10;
Query OK, 0 rows affected, 1 warning (0.02 sec)
```