[玩转 MySQL 之三]MySQL 用户及权限管理

作者：bangbang
原文链接：https://ld246.com/article/1537437418718
来源网站：链滴

<h2 id="引言">引言</h2>
数据库保存着应用程序日积夜累记录下来的数据资产，安全级别特别高，所以只能让授权的用户以访问，其他用户需一律拒绝。MySQL 是一个多用户数据库，拥有功能强大的访问控制系统，可以不同的用户指定不同的权限。 小编一直对 MySQL 的用户及权限管理都是一知半解，存有疑问，具体问如下:
1.MySQL 如何认证一个用户？
小编认为"用户认证"就是为了解决一个问题:你是谁？。在国内，中国公民要证明他是谁，只要拿出身证就可以，因为身份证上的照片，姓名，家庭住址，性别，出生年月，身份证号码等信息，是中国政为了说明你就是中国大地上某个地方的某某某而制定的。那么在 MySQL Server 中，一个用户是如何证的？
2.MySQL 的权限分哪几种及存储在什么地方？
小编认为"MySQL 权限"是为了解决一个问题: 你能在 MySQL Server 内干哪些事情？就好比图书馆样，只有办了卡的人才允许进入，不同的卡可以进入不同的图书馆区域，可以做不同的事情，即拥有一样的权限，那么 MySQL 的权限有哪些？并且这些权限存储在哪里？
3. MySQL 是如何控制用户访问的？
继续使用图书馆的栗子，当你要进图书馆的时候，需要刷卡或者与管理员沟通，如果无效，那么将会现谢绝参阅的礼貌回复；假如你有权限进入图书馆，但是你没有借书的权利，那么在你借书的时候，借书失败。在 MySQL Server 中， 一个用户想要对 MySQL Server 进行操作，MySQL Server 是如控制用户行为的？
<h2 id="一-MySQL用户认证">一、MySQL 用户认证</h2>
MySQL 的用户认证形式是: 用户名 + 主机。比如 test@127.0.0.1 和 test@192.168.10.10 是不样的用户。就好比现实中的牛家村的张三和马家村的张三是分别两个人一样。MySQL 中的权限分配是分配到用户 + 主机的实体上。MySQL 的主机信息可以是本地(localhost)，某个 IP，某个 IP 段，及任何地方等，即用户的地址可以限制到某个具体的 IP，或者某个 IP 范围，或者任意地方。MySQL 用户分为普通用户和 root 用户。root 用户是超级管理员，拥有所有权限，普通用户只拥有被授予的种权限。
<h2 id="二-MySQL的权限分类及存储">二、MySQL 的权限分类及存储</h2>
1.MySQL 用户权限层级

全局层级：全局权限适用于一个给定 MySQL Server 中的所有数据库，这些权限存储在 mysql.user 表中。

<pre><code class="highlight-chroma">GRANT ALL ON *.* TO 'user'@'host'; # *.* 表示数据库库的所有库和表，对应权限存储在mysql user表中

</code></pre>

数据库层级：数据库权限适用于一个给定数据库中的所有目标，这些权限存储在 mysql.db 表中

<pre><code class="highlight-chroma">GRANT ALL ON mydb.* TO 'user'@'host'; #mydb.* 表示mysql数据库下的所有表，对应权限储在mysql.db表中

</code></pre>

表层级：表权限适用于一个给定表中的所有列，这些权限存储在 mysql.tables_priv 表中。

<pre><code class="highlight-chroma">GRANT ALL ON mydb.mytable TO 'user'@'host'; #mydb.mytable 表示mysql数据库下的mytable表，对应权限存储在mysql.tables_priv表

</code></pre>

列层级：列权限使用于一个给定表中的单一列，这些权限存储在 mysql.columns_priv 表中。</l
>

<pre><code class="highlight-chroma"><span class="highlight
cl">GRANT ALL (col1， col2， col3) ON mydb.mytable TO 'user'@'host'; #mydb.mytable 表
mysql数据库下的mytable表， col1, col2，col3表示mytable表中的列名

</code></pre>

子程序层级：CREATE ROUTINE、ALTER ROUTINE、EXECUTE 和 GRANT 权限适用于已存储
子程序。这些权限可以被授予为全局层级和数据库层级。而且，除了 CREATE ROUTINE 外，这些权
可以被授予子程序层级，并存储在 mysql.procs_priv 表中。

<pre><code class="highlight-chroma"><span class="highlight
cl">GRANT EXECUTE ON PROCEDURE mydb.myproc TO 'user'@'host'; #mydb.mytable 表示m
sql数据库下的mytable表，PROCEDUR表示存储过程

</code></pre>
2. MySQL 权限简单分类

数据权限分为：库、表和字段三种级别
管理权限主要是管理员要使用到的权限，包括：数据库创建，临时表创建
主从部署、进程管理等
程序权限主要是触发器、存储过程、函数等权限。

3. MySQL 权限详情

<blockquote>
第一列表示可以在 grant 命令中制定的权限
第二列对应着几张权限表(user,db,tables_priv, columns_priv, procs_priv)中的列
第三列表示权限的作用范围，其中 Global（Server administration）对应 mysql.user 表，Database
对应 mysql.db 表，Tables 对应 mysql.tables_priv 表，Columns 对应 mysql.columns_priv 表，St
red routines 对应 mysql.procs_priv 表。
</blockquote>
MYSQL 的权限如何分布，就是针对表可以设置什么权限，针对列可以设置什么权限等等，这个
以从官方文档中的一个表来说明：
<table>
<thead>
<tr>
<th>权限分布</th>
<th>可能设置的权限</th>
</tr>
</thead>
<tbody>
<tr>
<td>表权限</td>
<td>Select, Insert, Update, Delete, Create, Drop, Grant, References, Index, Alter</td>
</tr>
<tr>
<td>列权限</td>

<td>Select, Insert, Update, References</td>
</tr>
<tr>
<td>程序权限</td>
<td>Execute, Alter Routine, Grant</td>
</tr>
</tbody>
</table>
<h2 id="三--MySQL访问控制">三、 MySQL 访问控制</h2>
MySQL 访问控制分为两个阶段:

用户连接检查阶段
执行 SQL 语句时检查阶段

<h2 id="1-用户连接时的检查">1、用户连接时的检查</h2>
1）当用户连接时，MySQL 服务器首先从 user 表里匹配 host, user, password，匹配不到则拒该连接
2）接着检查 user 表的 max_connections 和 max_user_connections，如果超过上限则拒绝连接<b>
3）检查 user 表的 SSL 安全连接，如果有配置 SSL，则需确认用户提供的证书是否合法只有上面 3 检查都通过后，服务器才建立连接，连接建立后，当用户执行 SQL 语句时，需要做 SQL 语句执行检。
<h2 id="2-执行SQL语句时的检查">2、执行 SQL 语句时的检查</h2>
1）从 user 表里检查 max_questions 和 max_updates，如果超过上限则拒绝执行 SQL 下面几是进行权限检查：
2）首先检查 user 表，看是否具有相应的全局性权限，如果有，则执行，没有则继续下一步检查
3）接着到 db 表，看是否具有数据库级别的权限，如果有，则执行，没有则继续下一步检查
4）最后到 tables_priv, columns_priv, procs_priv 表里查看是否具有相应对象的权限从以上的过程们可以知道，MySQL 检查权限是一个比较复杂的过程，所以为了提高性能，MySQL 的启动时就会这 5 张权限表加载到内存。
<h2 id="四-权限表字段详解">四、权限表字段详解</h2>
1.user 表 user 表的权限是基于服务器范围的所有权限，比如用户拥有服器中所有数据库的 select 权限那么在 user 表中的 Select_priv 列为 Y,如果用户单单只拥有某个一数库的 select 权限那么 user 表中的 Select_priv 为 N,会在 DB 表中记录一条信息在 DB 表中的 select_riv 为 Y。
<pre><code class="highlight-chroma">desc mysql.user;

</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile com/e/863482f6f924409a9d706c626a36f23b.png?imageView2/2/interlace/1/format/jpg"></
>
2.db 表如果授予一个用户单独某个数据库的权限，就会在 db 表中记录一相关信息。
<pre><code class="highlight-chroma">desc mysql.db;

</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile com/e/0f977f6d728643a5b278413b1074d19c.png?imageView2/2/interlace/1/format/jpg"><
p>
3.tables_priv 表
<pre><code class="highlight-chroma">desc mysql.tables_priv;

</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/e/4d15f728b7684c2dbb3beef7fd2d6457.png?imageView2/2/interlace/1/format/jpg"><
p>
<blockquote>
上面的 Column_priv 比较奇怪，因为照理说 tables_priv 只显示表级别的权限，列级别的权限
该在 columns_priv 里显示才对。后来查了资料才知道，原来这是为了提高权限检查时的性能，试想
下，权限检查时，如果发现 tables_priv.Column_priv 为空，就不需要再检查 columns_priv 表了，
种情况在现实中往往占大多数。
</blockquote>
4. columns_priv 表
<pre><code class="highlight-chroma"><span class="highlight
cl">desc mysql.columns_priv;

</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/e/3fe0090cbf7d471dbeeab9999bad4922.png?imageView2/2/interlace/1/format/jpg"><
p>
5. procs_priv 表
<pre><code class="highlight-chroma"><span class="highlight
cl">desc mysql.procs_priv;

</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/e/5c98d673c92e4179b53cf6a4bade72d3.png?imageView2/2/interlace/1/format/jpg"><
p>
<h2 id="五-用户管理实践">五、用户管理实践</h2>
1.用户创建

通过 create user 语句创建用户

在执行 CREATE USER 或 CRANT 语句后，MySQL 服务器会修改相应的用户权限表，添加或修
用户及权限。
<pre><code class="highlight-chroma"><span class="highlight
cl"> create user 'USERNAME'@'HOST' identified by 'PASSWORD';

</code></pre>
<blockquote>
HOST 的形式：

IP 地址，如 172.16.16.1；
主机名，如 localhost；
网络地址，如 172.16.0.04. 通配符，如 %：匹配任意字符_：匹配任意单个字符如 172.16.16._(
许 172.16.16.1-172.16.16.9)

</blockquote>
eg:
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> CREATE USER 'jeffrey'@'localhost' IDENTIFIED BY 'mypass';
Query OK, 0 rows
ffected (0.00 sec)

</code></pre>

通过 grant 语句创建新用户

GRANT 语句是添加新用户并授权它们访问 MySQL 对象的首选方法，其语法格式为：
<pre><code class="highlight-chroma"><span class="highlight
cl">grant all on DB_NAME.TABLE_NAME to 'USERNAME'@'HOST' identified by 'PASSWORD';

</code></pre>
<blockquote>
HOST 的表现形式和 create user 一样
</blockquote>
eg：
<pre><code class="highlight-chroma"><span class="highlight
cl"># 用户 testUser对所有的数据有查询和更新权限，并授于对所有数据表的SELECT和UPDATE权限
mysql> GRANT
SELECT,UPDATE ON *.* TO 'testUser'@'localhost' IDENTIFIED BY 'testpwd';
Query OK, 0 rows
ffected (0.00 sec)

</code></pre>

创建 root 用户

<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'root' WIT
 GRANT OPTION;
mysql> flush pr
vileges;

</code></pre>
2). 创建一个基本的增删改查用户
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> GRANT UPDATE, DELETE, INSERT, SELECT ON *.* TO 'test'@'%' identified by 'te
t' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER
HOUR 0 MAX_USER_CONNECTIONS 0;
mysql> flush pr
vileges;
MAX_QUERIES_
ER_HOUR，MAX_CONNECTIONS_PER_HOUR，MAX_UPDATES_PER_HOUR设置为0表示不限制

</code></pre>
3). 创建数据库基本的增删改查用户
<pre><code class="language-mysql highlight-chroma"><span
class="highlight-cl">mysql&
t; GRANT<s
an class="highlight-w"> SELECT<span class="hi
hlight-p">, INSERT<
span>, <span class=
highlight-k">UPDATE,<span class="highlight-w
> DELETE,<
pan class="highlight-w"> SHOW<span class="h
ghlight-w"> VIEW,<
span> EXECUTE<sp
n class="highlight-w"> ON<span class="highlig

t-w"> `db_name</s
an>`.<span class="h
ghlight-o">* TO</s
an> 'test'<span cla
s="highlight-o">@'%'<span class="highlight-w
> identified </span
by <span class="hi
hlight-s1">'test';
<span cla
s="highlight-w">
</code></pre>
4). 授予数据库名以 db 开头的数据库的权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> GRANT SELECT, INSERT, UPDATE, DELETE, SHOW VIEW, EXECUTE ON `db%`.*
O 'perform'@'%';
mysql> flush pr
vileges;

</code></pre>
5). 创建备份用户权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> GRANT SELECT,EVENT,SHOW DATABASES,LOCK TABLES,SUPER,REPLICATION
LIENT ON *.* TO 'backup'@'localhost' identified by 'backup';
mysql> flush pr
vileges;

</code></pre>
6). 备份恢复用户权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> GRANT INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER,CREATE VIEW,
SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER ON *.* TO 'restore'@'local
ost' identified by '123456';
mysql> flush pr
vileges;

</code></pre>
7). 复制用户权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> GRANT PROCESS, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'repl'
'%' IDENTIFIED BY '123456';
mysql> flush pr
vileges;

</code></pre>
2.用户删除
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> drop user 'USERNAME'@'HOST';

</code></pre>

<pre><code class="highlight-chroma"># 删除MySQL默认的无用账户;
mysql> drop user 'root'@'localhost.localdomain';

删除MySQL默的无用账户;
mysql> drop user 'root'@'127.0.0.1';

</code></pre>
3. 更改用户名
<pre><code class="highlight-chroma">mysql> rename user OLD_NAME to NEW_NAME;

</code></pre>
4. 修改用户密码

通过 mysqladmin 工具

<pre><code class="highlight-chroma"># 给root@localhost用户登录mysql设置密码为"redhat";
$ mysqladmin -uoot -h localhost password "redhat"

# 修改root@localost用户登录mysql数据库的密码;
$ mysqladmin -uoot -h localhost password "new passwd" -p "old passwd"

</code></pre>

通过直接修改 mysql.user 表的用户记录

<pre><code class="highlight-chroma"># MySQL 5.6
mysql> update mysql.user set password=PASSWORD('redhat') where user='root';
mysql> flush privileges;

MySQL 5.7
mysql> update mysql.user set authentication_string=PASSWORD('redhat') where user='root';
mysql> flush privileges;

</code></pre>

set password 语句

<pre><code class="highlight-chroma">mysql> set password for 'root'@'localhost'=PASSWORD('redhat');
mysql> flush pr

vileges;

</code></pre>

ALTER USER 语句(MYSQL5.7 版本)

<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> use mysql
mysql> alter u
er root@'localhost' identified by '123456';
mysql> flush pr
vileges;

</code></pre>
<h2 id="六-MySQL管理员密码找回">六、MySQL 管理员密码找回</h2>
1.修改配置文件，跳过授权表在配置文件中[mysqld]字段添加 skip-grant-t
bles 指令
<pre><code class="highlight-chroma"><span class="highlight
cl">$ cat /etc/my.cnf
[mysqld]
skip-grant-tables

</code></pre>
2. 重启 MySQL Server
<pre><code class="highlight-chroma"><span class="highlight
cl">service mysqld restart

</code></pre>
3. 给 root 用户登录 mysql 设置密码为 helloWORD 并以加密方式
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> use mysql;
MySQL5.6
mysql> update
user set password=PASSWORD('helloWORD') where user='root';
MySQL5.7
mysql> update
mysql.user set authentication_string=PASSWORD('helloWORD') where user='root';
mysql> flush p
ivileges;

</code></pre>
4.修改配置文件，注释刚才添加的配置项
<pre><code class="highlight-chroma"><span class="highlight
cl">$ cat /etc/my.cnf
[mysqld]
#skip-grant-tables

</code></pre>
5.重启 MySQL Server
<pre><code class="highlight-chroma"><span class="highlight
cl">service mysqld restart

</code></pre>
<h2 id="七--MySQL-权限管理实践">七、 MySQL 权限管理实践</h2>

账户权限信息被存储在 MySQL 数据库的几张权限表中，在 MySQL 启动时，服务器将这些数据表中权限信息的内容读入内存。其中 GRANT 和 REVOKE 语句所涉及的常用权限大致如下这些：CRETE、DROP、SELECT、INSERT、UPDATE、DELETE、INDEX、ALTER、CREATE、ROUTINE、FILE 等，还有一个特殊的 proxy 权限，是用来赋予某个用户具有给他人赋予权限的权限。
1. grant 所有权限
<pre><code class="highlight-chroma">mysql> grant all privileges on *.* to 'USERNAME'@'HOST';
mysql> flush privileges;

</code></pre>
2. grant super 权限在.上(super 权限可以对全局变量更改)；
<pre><code class="highlight-chroma">mysql> grant super on *.* to 'USERNAME'@'HOST';
mysql> flush privileges;

</code></pre>
3. grant 某个库下所有表的所有权限
<pre><code class="highlight-chroma">mysql> grant all privileges on DB_NAME.* to 'USERNAME'@'HOST';
mysql> flush privileges;

</code></pre>
4. grant 某个库下所有表的 select 权限
<pre><code class="highlight-chroma">mysql>grant select on DB_NAME.* to 'USERNAME'@'HOST';
mysql> flush privileges;

</code></pre>
5. grant 某个库下某个表的 insert 权限
<pre><code class="highlight-chroma">mysql> grant insert on DB_NAME.TABLE_NAME to 'USERNAME'@'HOST';
mysql> flush privileges;

</code></pre>
6. grant 某个库下某个表的 update 权限
<pre><code class="highlight-chroma">mysql>grant update on DB_NAME.TABLE_NAME to 'USERNAME'@'HOST';
mysql> flush privileges;

</code></pre>
7. grant 某个库下某个表的某个字段 update 权限
<pre><code class="highlight-chroma">mysql> grant update(COLUMN_NAME) on DB_NAME.TABLE_NAME to 'USERNAME'@HOST';
mysql> flush privileges;

</code></pre>
8.通过 GRANT 语句中的 USAGE 权限，可以创建账户而不授予任何权限</
>
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant usage on *.* to 'USERNAME'@'HOST';
mysql> flush p
ivileges;

</code></pre>
9. grant 创建、修改、删除 MySQL 数据表结构权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant create on testdb.* to developer@'192.168.0.%';
mysql> grant a
ter on testdb.* to developer@'192.168.0.%';
mysql> grant d
op on testdb.* to developer@'192.168.0.%';
mysql> flush p
ivileges;

</code></pre>
10. grant 操作 MySQL 外键权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant references on testdb.* to developer@'192.168.0.%';
mysql> flush p
ivileges;

</code></pre>
11. grant 操作 MySQL 临时表权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant create temporary tables on testdb.* to developer@'192.168.0.%';
mysql> flush p
ivileges;

</code></pre>
12. grant 操作 MySQL 索引权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant index on testdb.* to developer@'192.168.0.%';
mysql> flush p
ivileges;

</code></pre>
13.grant 操作 MySQL 视图、查看视图源代码权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant create view on testdb.* to developer@'192.168.0.%';
mysql> grant s
ow view on testdb.* to developer@'192.168.0.%';
mysql> flush pr
vileges;

</code></pre>
14. grant 操作 MySQL 存储过程、存储函数权限
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant create routine on testdb.* to developer@'192.168.0.%';
mysql> grant a

ter routine on testdb.* to developer@'192.168.0.%';
mysql> grant e
ecute on testdb.* to developer@'192.168.0.%';
mysql> flush pr
vileges;

</code></pre>
15.PROXY 特殊权限
如果想让某个用户具有给他人赋予权限的能力，那么就需要 proxy 权限了。当你给一个用户赋予 all
限之后，你查看 mysql.user 表会发现 Grant_priv 字段还是为 N，表示其没有给他人赋予权限的权限

我们可以查看一下系统默认的超级管理员权限：
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> show grants for 'root'@'localhost';
+----------------
--+
| Grants for root@
ocalhost |
+----------------
--+
| GRANT ALL PRIV
LEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
| GRANT PROXY
N ''@'' TO 'root'@'localhost' WITH GRANT OPTION |
+----------------
--+
2 rows in set (0.00
sec)

</code></pre>
可以看到其本身有 PROXY 权限，并且这个语句跟一般授权语句还不太一样。所以如果想让一个
程用户有给他人赋予权限的能力，就需要给此用户 PROXY 权限，如下：
<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> grant all on *.* to 'test'@'%' identified by 'helloWORD';
mysql> GRANT
PROXY ON ''@'' TO 'test'@'%' WITH GRANT OPTION;
mysql> flush pr
vileges;

</code></pre>
16. 查看用户的权限
<pre><code class="highlight-chroma"><span class="highlight
cl">Mysql> show grants for 'USERNAME'@'HOST';

</code></pre>
17. 移除用户权限
<pre><code class="highlight-chroma"><span class="highlight
cl"># 移除tom用户对于db.xsb的权限;
Mysql> revoke
all on db.xsb from 'tom'@'localhost';
刷新授权表;
Mysql> flush pr
vileges;

</code></pre>
<blockquote>
使用 REVOKE 收回权限之后，用户帐户的记录将从 db、host、tables_priv、columns_priv 表删除，但是用户帐号依然在 user 表中保存。
</blockquote>
<h2 id="八-MySQL-用户和权限管理经验">八、MySQL 用户和权限管理经验</h2>
1. 用户管理经验

1)、尽量使用 create user, grant 等语句，而不要直接修改权限表。

虽然 create user, grant 等语句底层也是修改权限表，和直接修改权限表的效果是一样的，但是对于非高手来说，采用封装好的语句肯定不会出错，而如果直接修改权限表，难免会漏掉某些表。而，修改完权限表之后，还需要执行 flush privileges 重新加载到内存，否则不会生效。

2). 把匿名用户删除掉。

匿名用户没有密码，不但不安全，还会产生一些莫名其妙的问题，强烈建议删除。
2. 权限管理经验

1)、只授予能满足需要的最小权限，防止用户干坏事。比如用户只是需要查询，那就只给 select 限就可以了，不要给用户赋予 update、insert 或者 delete 权限。
2)、创建用户的时候限制用户的登录主机，一般是限制成指定 IP 或者内网 IP 段。
3)、初始化数据库的时候删除没有密码的用户。安装完数据库的时候会自动创建一些用户，这些户默认没有密码。
4)、为每个用户设置满足密码复杂度的密码。
5)、定期清理不需要的用户，回收权限或者删除用户。

<h2 id="九-参考文献">九、参考文献</h2>
MySQL 之权限管理<b>
MySQL 用户和权限管理
MySQL 权限
探索权限表
MySQL 权限机制和权限存储
<h2 id="更多内容请关注公众号">更多内容请关注公众号</h2>