#第三周 任务名称：数据库系统表相关学习

- ***如何利用数据库的功能读写文件，需要什么样的条件才可以读写***

MySQL的几种读取文件方法，大致分为：

1.load_file()
2.load data infile()
3.system cat

load_file()和load data infile读取文件的方法为:新建一个表，读取文件为字符串形式插入表中，然后读出表中数据。

但是通常情况下有两个前提:

1.在拥有file权限的前提下

2.secure_file_priv不为NULL

secure_file_priv的值可以在这样查看

```
mysql> show global variables like "secure_file_priv";
+------------------+-----------------------+
| Variable_name    | Value                 |
+------------------+-----------------------+
| secure_file_priv | /var/lib/mysql-files/ |
+------------------+-----------------------+
1 row in set (0.00 sec)
```

在mysql 5.6.34版本以后 secure_file_priv的值默认为NULL。可以通过以下方式修改

windows下:

修改mysql.ini 文件，在[mysqld] 下添加条目: secure_file_priv =

保存，重启mysql。

Linux下:

在/etc/my.cnf的[mysqld]下面添加local-infile=0选项。

一、load_file()

首先在有读写权限的目录创建文档

```
[root@localhost tmp]# cd /var/lib/mysql-files/
[root@localhost mysql-files]# ls
[root@localhost mysql-files]# vim test.txt
[root@localhost mysql-files]# vim test2.txt
[root@localhost mysql-files]# vim test3.txt
[root@localhost mysql-files]#
[root@localhost mysql-files]#
[root@localhost mysql-files]# vim /etc/my.cnf
[root@localhost mysql-files]#
root@localhost:/var/lib/mysql-files                    1 / 4
```

运行mysql

sql命令如下:

```
create table user(cmd text);
```

```
insert into user(cmd) values (load_file('/var/lib/mysql-files/test.txt'));
```

```
select * from user;
```

```
mysql> insert into user(cmd) values (load_file('/var/lib/mysql-files/test.txt'));
Query OK, 1 row affected (0.00 sec)

mysql> select * from user;
+------------------------------+
| cmd                          |
+------------------------------+
| NULL                         |
| test
test
test
test
test
test
 |
+------------------------------+
2 rows in set (0.00 sec)
```

二、load data infile

其实load data infile和load_file()用法上没有什么区别,只是在注入过程中，往往会过滤掉load_file()这个函数，但是仍然有load data infile可以使用。

```
load data infile '/var/lib/mysql-files/test2.txt' into table user;
```

```
mysql> load data infile '/var/lib/mysql-files/test2.txt' into table user;
Query OK, 5 rows affected (0.00 sec)
Records: 5  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from user;
+------------------------------+
| cmd                          |
+------------------------------+
| NULL                         |
| test
test
test
test
test
test
 |
| test2                        |
| test2                        |
| test2                        |
| test2                        |
```

三、system cat

在mysql版本为5.x时,除了可以使用上两种方法外，还可以使用系统命令直接读取文件

```
mysql> system cat /var/lib/mysql-files/test3.txt
test3
test3
test3
test3
test3
test3
mysql>
```

注意：

1.此方法只能在本地读取，远程连接mysql时无法使用system。

2.无法越权读取。

- ***学习数据库系统表的功能，如何利用 sql 语句查询库名、表名、字段名、内容以及当前用户等基本信息，将学习过程中关键部分整理成报告***

1、查看用户

内置的mysql数据库中的user表存储了所有用户信息，可以通过以下语句查询：

`select user,host from mysql.user;`

```
mysql> select user,host from mysql.user;
+---------------+-----------+
| user          | host      |
+---------------+-----------+
| root          | %         |
| mysql.session | localhost |
| mysql.sys     | localhost |
+---------------+-----------+
3 rows in set (0.01 sec)
```

显示所有用户(不重复)

select distinct user from mysql.user;

```
mysql> select distinct user from mysql.user;
+---------------+
| user          |
+---------------+
| root          |
| mysql.session |
| mysql.sys     |
+---------------+
3 rows in set (0.00 sec)
```

查看当前登录用户：

select user();

或

SELECT CURRENT_USER();

```
mysql> select user();
+----------------+
| user()         |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)

mysql> select current_user();
+----------------+
| current_user() |
+----------------+
| root@%         |
+----------------+
1 row in set (0.00 sec)

mysql>
```

2、查看数据库名

在MySQL5.0版本后面。MySQL默认在数据库存放一个information_schema的数据库，在该库中需要记住三个表名，分别为SCHEMATA、TABLES、和COLUMNS。

SCHEMATA表存储该用户创建的所有数据库的库名，需要记住SCHEMATA_NAME字段，既是所有的数据库名

```
mysql> select * from information_schema.schemata;
+--------------+--------------------+----------------------------+------------------------+-------
---+
| CATALOG_NAME | SCHEMA_NAME        | DEFAULT_CHARACTER_SET_NAME | DEFAULT_COLLATION_NAME | SQL_PA
TH |
+--------------+--------------------+----------------------------+------------------------+-------
---+
| def          | information_schema | utf8                       | utf8_general_ci        | NULL
   |
| def          | mysql              | latin1                     | latin1_swedish_ci      | NULL
   |
| def          | performance_schema | utf8                       | utf8_general_ci        | NULL
   |
| def          | sys                | utf8                       | utf8_general_ci        | NULL
   |
| def          | test_db            | utf8                       | utf8_general_ci        | NULL
   |
+--------------+--------------------+----------------------------+------------------------+-------
---+
5 rows in set (0.00 sec)
```

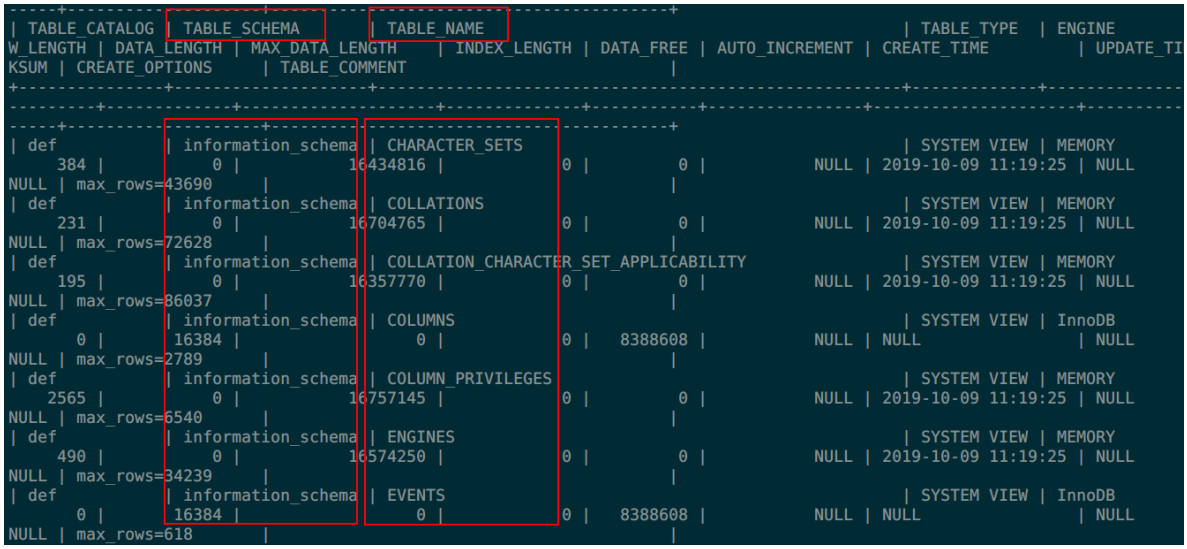
获取当前数据库名

`select database();`

```
mysql> select database();
+------------+
| database() |
+------------+
| mysql      |
+------------+
1 row in set (0.00 sec)
```

3、查看表名

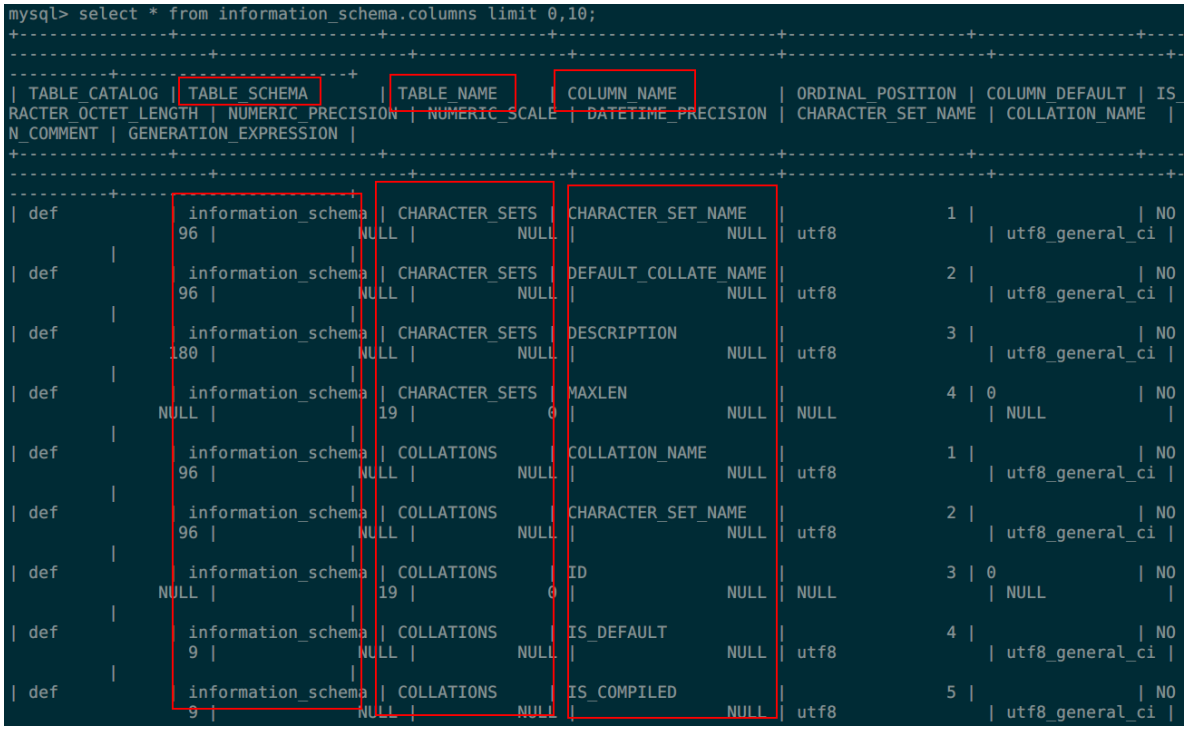
TABLES表存储该用户创建的所有数据库的库名和表名，需记住TABLE_SCHEMA和TABLE_NAME字段。



```
| TABLE_CATALOG | TABLE_SCHEMA       | TABLE_NAME                                                    | TABLE_TYPE  | ENGINE
W_LENGTH | DATA_LENGTH | MAX_DATA_LENGTH    | INDEX_LENGTH | DATA_FREE | AUTO_INCREMENT | CREATE_TIME         | UPDATE_TI
KSUM | CREATE_OPTIONS     | TABLE_COMMENT                                                 |
| def           | information_schema | CHARACTER_SETS                                                | SYSTEM VIEW | MEMORY
     384 |           0 |           16434816 |            0 |         0 |           NULL | 2019-10-09 11:19:25 | NULL
NULL | max_rows=43690     |                                                               |
| def           | information_schema | COLLATIONS                                                    | SYSTEM VIEW | MEMORY
     231 |           0 |           16704765 |            0 |         0 |           NULL | 2019-10-09 11:19:25 | NULL
NULL | max_rows=72628     |                                                               |
| def           | information_schema | COLLATION_CHARACTER_SET_APPLICABILITY                         | SYSTEM VIEW | MEMORY
     195 |           0 |           16357770 |            0 |         0 |           NULL | 2019-10-09 11:19:25 | NULL
NULL | max_rows=86037     |                                                               |
| def           | information_schema | COLUMNS                                                       | SYSTEM VIEW | InnoDB
       0 |       16384 |                0 |            0 |   8388608 |           NULL | NULL                | NULL
NULL | max_rows=2789      |                                                               |
| def           | information_schema | COLUMN_PRIVILEGES                                             | SYSTEM VIEW | MEMORY
    2565 |           0 |           16757145 |            0 |         0 |           NULL | 2019-10-09 11:19:25 | NULL
NULL | max_rows=6540      |                                                               |
| def           | information_schema | ENGINES                                                       | SYSTEM VIEW | MEMORY
     490 |           0 |           16574250 |            0 |         0 |           NULL | 2019-10-09 11:19:25 | NULL
NULL | max_rows=34239     |                                                               |
| def           | information_schema | EVENTS                                                        | SYSTEM VIEW | InnoDB
       0 |       16384 |                0 |            0 |   8388608 |           NULL | NULL                | NULL
NULL | max_rows=618       |                                                               |
```

4、查看字段名

COLUMNS表存储该用户所有数据库的库名、表名和字段名，需记住TABLE_SCHEMA、TABLE_NA ME和COLUMN_NAME字段。



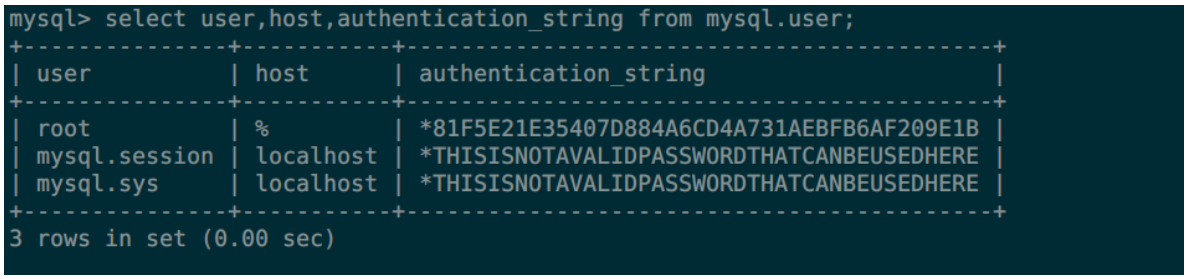
```
mysql> select * from information_schema.columns limit 0,10;
| TABLE_CATALOG | TABLE_SCHEMA       | TABLE_NAME     | COLUMN_NAME          | ORDINAL_POSITION | COLUMN_DEFAULT | IS_
RACTER_OCTET_LENGTH | NUMERIC_PRECISION | NUMERIC_SCALE | DATETIME_PRECISION | CHARACTER_SET_NAME | COLLATION_NAME  |
N_COMMENT | GENERATION_EXPRESSION |
| def           | information_schema | CHARACTER_SETS | CHARACTER_SET_NAME    |                1 |                | NO
                96 |                NULL |          NULL |                NULL | utf8               | utf8_general_ci |
           |                       |
| def           | information_schema | CHARACTER_SETS | DEFAULT_COLLATE_NAME  |                2 |                | NO
                96 |                NULL |          NULL |                NULL | utf8               | utf8_general_ci |
           |                       |
| def           | information_schema | CHARACTER_SETS | DESCRIPTION           |                3 |                | NO
               180 |                NULL |          NULL |                NULL | utf8               | utf8_general_ci |
           |                       |
| def           | information_schema | CHARACTER_SETS | MAXLEN                |                4 | 0              | NO
              NULL |                  19 |             0 |                NULL | NULL               | NULL            |
           |                       |
| def           | information_schema | COLLATIONS     | COLLATION_NAME        |                1 |                | NO
                96 |                NULL |          NULL |                NULL | utf8               | utf8_general_ci |
           |                       |
| def           | information_schema | COLLATIONS     | CHARACTER_SET_NAME    |                2 |                | NO
                96 |                NULL |          NULL |                NULL | utf8               | utf8_general_ci |
           |                       |
| def           | information_schema | COLLATIONS     | ID                    |                3 | 0              | NO
              NULL |                  19 |             0 |                NULL | NULL               | NULL            |
           |                       |
| def           | information_schema | COLLATIONS     | IS_DEFAULT            |                4 |                | NO
                 9 |                NULL |          NULL |                NULL | utf8               | utf8_general_ci |
           |                       |
| def           | information_schema | COLLATIONS     | IS_COMPILED           |                5 |                | NO
                 9 |                NULL |          NULL |                NULL | utf8               | utf8_general_ci |
```

扩展学习：尝试查询出用户的 hash，并使用 hashcat 来对获取的 hash 进行暴力破解

1、查询用户的hash

select user,host,authentication_string from mysql.user;

在MySQL 5.7 password字段已从mysql.user表中删除，新的字段名是“authenticalion_string”。

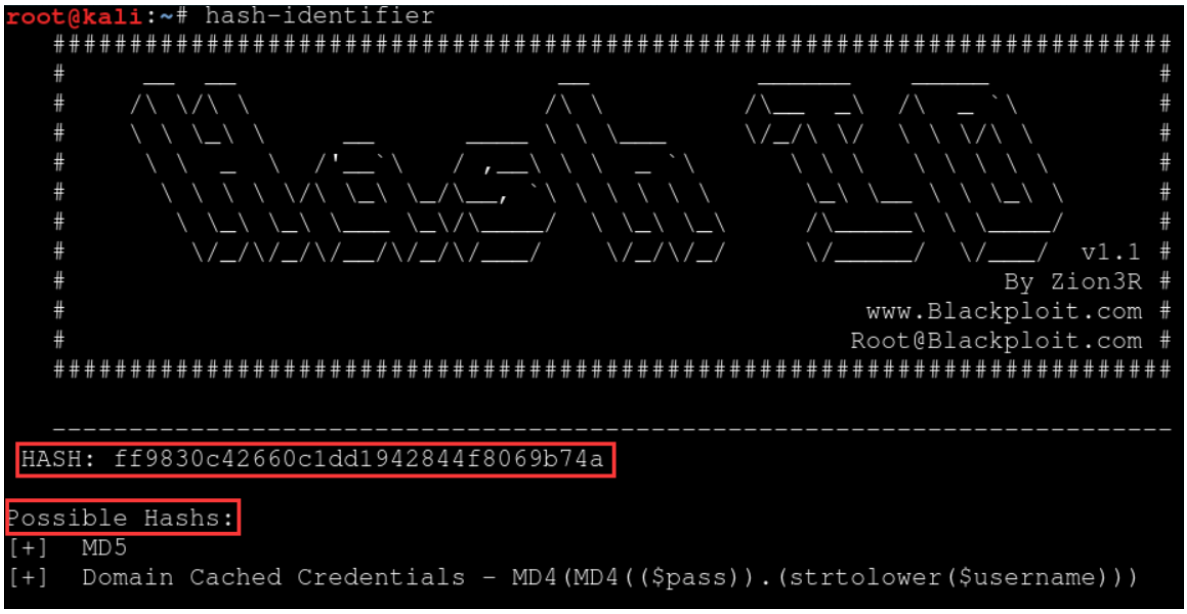


```
mysql> select user,host,authentication_string from mysql.user;
+---------------+-----------+-------------------------------------------+
| user          | host      | authentication_string                     |
+---------------+-----------+-------------------------------------------+
| root          | %         | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| mysql.session | localhost | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| mysql.sys     | localhost | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
+---------------+-----------+-------------------------------------------+
3 rows in set (0.00 sec)
```

2、识别hash类型

先用下面的脚本[kali自带]大致识别下,可能不太准,不过粗略的看一眼还是可以的,另外,谷歌上也有很多在线的hash类型识别站,可自行去尝试。

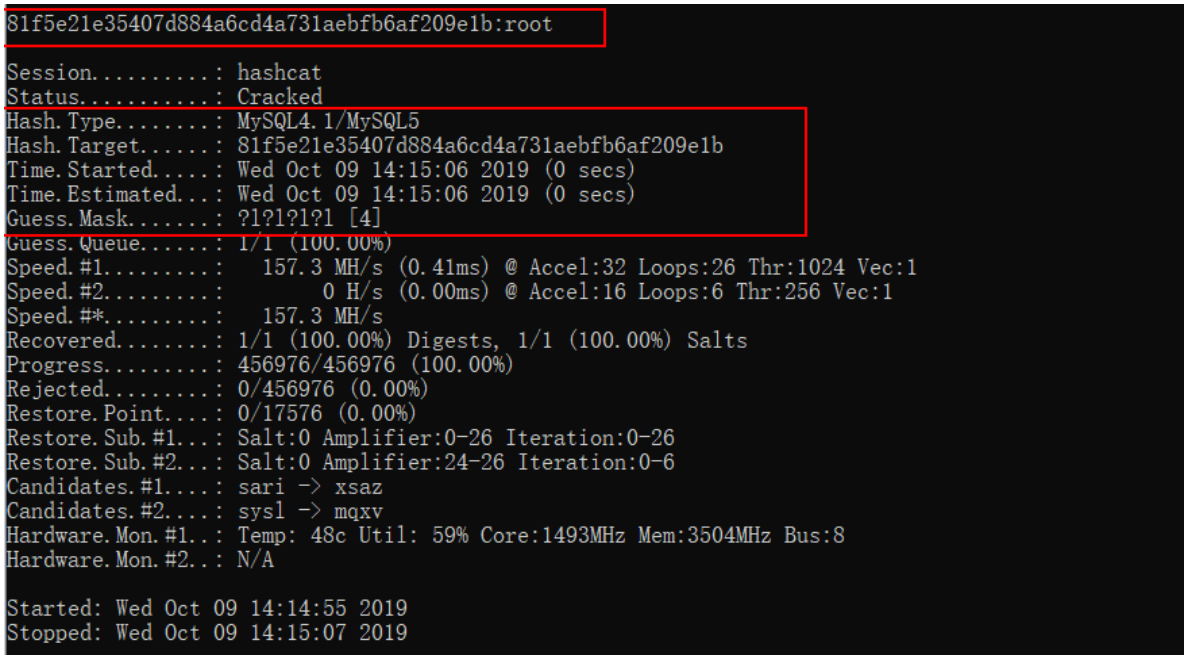
`hash-identifier`



3、破解 mysql 4/5.x 数据库用户hash

[记得实际破解的时候把hash开头的*去掉,要不然可能识别不出来]:

`hashcat64.exe --force -a 3 -m 300 81F5E21E35407D884A6CD4A731AEBFB6AF209E1B ?l?l?l?l`



参考链接:

https://klionsec.github.io/2017/04/26/use-hashcat-crack-hash/

https://xz.aliyun.com/t/4008