Mysql报错注入之floor(rand(0)*2)报错原理探究

N0r4h 2020-04-29 12:21:49 642409

一、简述

floor报错注入是利用 select count(*),(floor(rand(0)*2)) x from users group by x这个相对固定的语句格式，导致的数据库报错。实际利用中通过 concat 函数，连接注入语句与 floor(rand(0)*2)函数，就实现了注入结果与报错信息回显的注入方式。具体利用本文不做阐述，本文重点探究该语句报错的原因，要理解该语句的报错原因，首先大家需要理解如下的关键函数的作用： count() 、group by 、floor()、rand()。

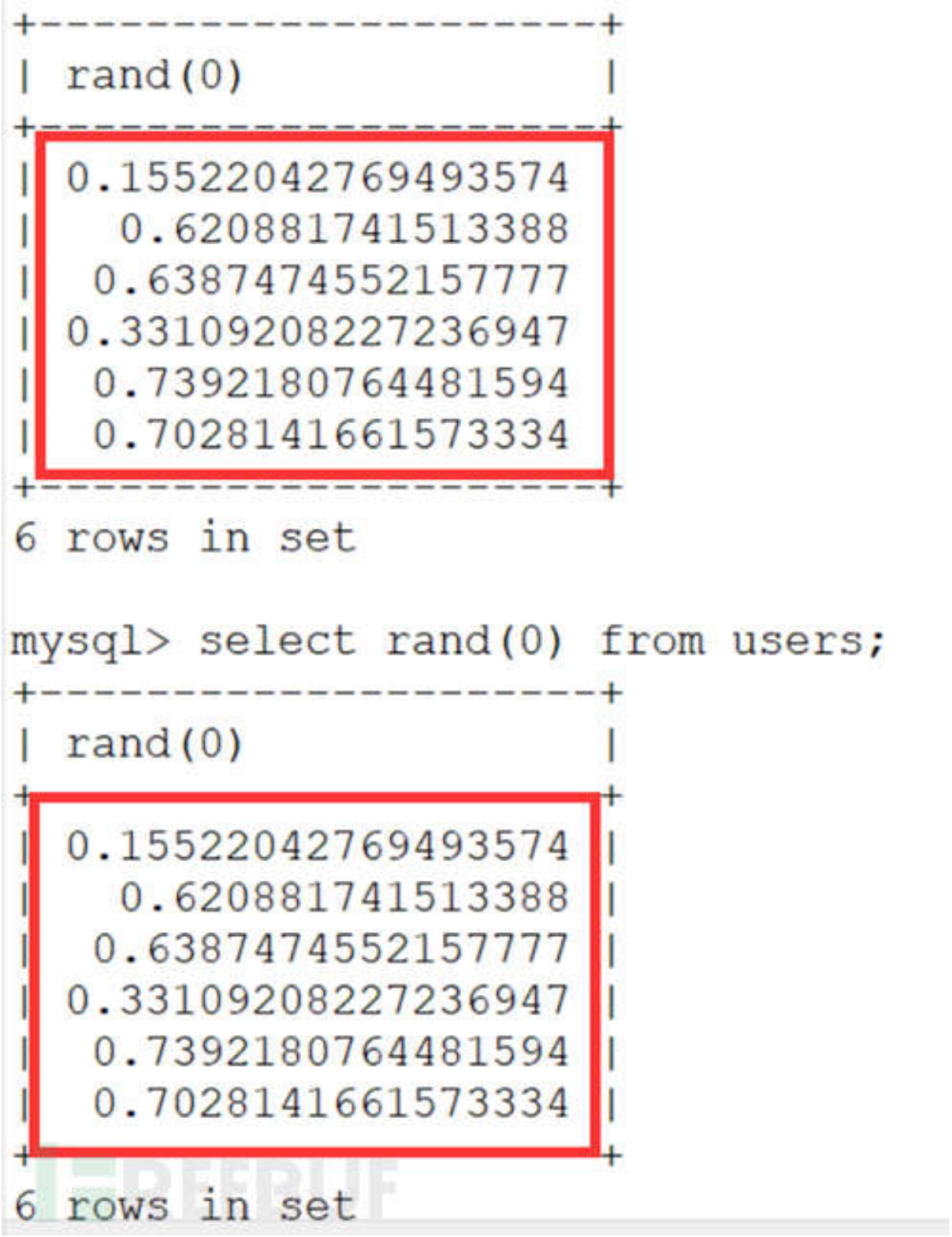
二、关键函数说明

1.理解rand函数

rand() 是一个随机函数，通过一个固定的随机数的种子0之后，可以形成固定的伪随机序列。结果如下图所示：

可见，直接使用rand函数每次产生的数都不同，但是当提供了一个固定的随机数的种子0之后：

这样每次产生的值都是一样的。也可以称之为伪随机（产生的数据都是可预知的）。
查看多个数据看一下。（users是一个有6行数据的表）



这样第一次产生的随机数和第二次完全一样，也就是可以预测的。

那么floor报错注入利用的时候rand（0）*2为什么要乘以 2 呢？这就要配合floor 函数来说了。

2.理解floor(rand(0)*2)函数

floor() 函数的作用就是返回小于等于括号内该值的最大整数，也就是取整。

floor(rand(0)*2）就是对rand(0)产生的随机序列诚意2后的结果，再进行取整。得到伪随机序列为如下图所示：

```
mysql> select floor(rand(0)*2) from users;
+------------------+
| floor(rand(0)*2) |
+------------------+
|                0 |
|                1 |
|                1 |
|                0 |
|                1 |
|                1 |
+------------------+
6 rows in set
```

因为使用了固定的随机数种子0，他每次产生的随机数列都是相同的0 1 1 0 1 1的顺序。

3.group by 函数

group by 主要用来对数据进行分组（相同的分为一组）。

例如建立如下表进行实验

```
mysql> select * from users;
+---------+-----------+
| user_id | user_name |
+---------+-----------+
|       1 | zhangsan  |
|       2 | lisi      |
|       3 | wangwu    |
|       4 | zhangsan  |
|       5 | lisi      |
|       6 | zhangsan  |
+---------+-----------+
6 rows in set
```

通过如下语句进行查询。（这里在a和x之前缺省了as，作用为用a和x代替原有的字段显示），显示的结果如下图所示：

```
mysql> select user_id a,user_name x from users ;
+---+----------+
| a | x        |
+---+----------+
| 1 | zhangsan |
| 2 | lisi     |
| 3 | wangwu   |
| 4 | zhangsan |
| 5 | lisi     |
| 6 | zhangsan |
+---+----------+
6 rows in set
```

但通过group by进行分组排序是，结果会进行分组，相同名字为合并。如下图所示

注意：最后x这列中显示的每一类只有一次，前面的a的是第一次出现的id值

```
mysql> select user_id a,user_name x from users group by x;
+---+----------+
| a | x        |
+---+----------+
| 2 | lisi     |
| 3 | wangwu   |
| 1 | zhangsan |
+---+----------+
3 rows in set
```

4.理解count（*）函数

count（*）统计结果的记录数。

这里与group by结合使用看一下：

```
mysql> select user_name a, count(*) x from users group by a;
+----------+---+
| a        | x |
+----------+---+
| lisi     | 2 |
| wangwu   | 1 |
| zhangsan | 3 |
+----------+---+
3 rows in set
```

这里就是对a中的重复性的数据进行了整合，然后计数，后面的x就是每一类的数量。也就是lisi有2个，wangwu有1个，zhangsan有3个。注意显示同样是按照ascii排序。

三、报错原因分析

大家已经了解，当执行如下语句时，就会产生一个报错。如下图所示

select count(*),floor(rand(0)*2) x from users group by x;

```
mysql> select count(*),floor(rand(0)*2) x from users group by x;
1062 - Duplicate entry '1' for key 'group_key'
```

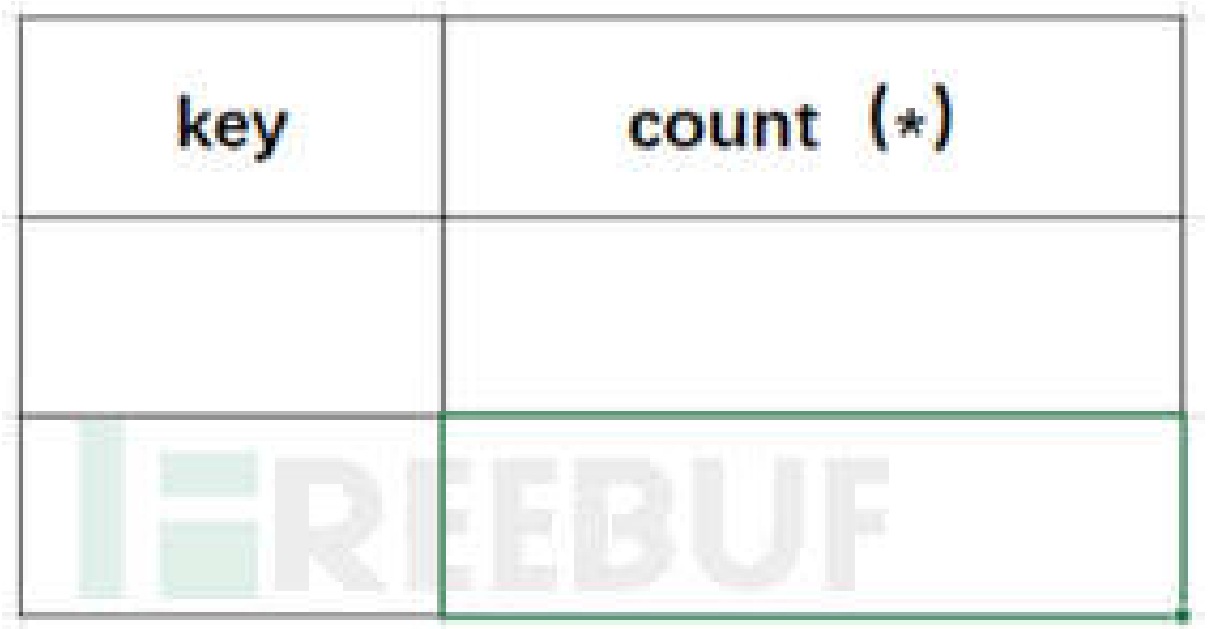
根据前面函数的理解，这句话本义就是统计后面产生随机数的种类并计算每种数量。原本执行结果一共6行数据，产生的随机序列应该为0 1 1 0 1 1 ，按照语句的含义，统计如果应该是：0是2个，1是4个，但是此处却产生了报错？这是为什么呢？下面来分析一下。

这里最关键的及时要理解group by函数的工作过程。group by key 在执行时循环读取数据的每一行，将结果保存于临时表中。读取每一行的key时，如果key存在于临时表中，则更新临时表中的数据（更新数据时，不再计算rand值）；如果该key不存在于临时表中，则在临时表中插入key所在行的数据。（插入数据时，会再计算rand值）

如果此时临时表只有key为1的行不存在key为0的行，那么数据库要将该条记录插入临时表，由于是随机数，插时又要计算一下随机值，此时 floor(random(0)*2)结果可能为1，就会导致插入时冲突而报错。即检测时和插入时两次计算了随机数的值。

具体报错原因可以通过下列过程展示：

mysql执行结果，会产生 011011 这个序列，group by时，会建立空虚拟表如下图，然后从sql语句执行结果序列（011011）读取数据并插入虚表:



（1）虚表写入第一条记录，执行floor(rand(0)*2)，发现结果为0(此时为第一次计算)

操作	key	floor（rand（0）*2）	count（*）
取第一条记录		0	

（2）查询虚拟表，发现0的键值不存在，则插入新的键值的时候floor(rand(0)*2)会被再计算一次，结果为1(此时为第二次计算)，插入虚表，第一条记录插入完毕，结果为1。如下图:

操作	key	floor（rand（0）*2）	count（*）
取第一条记录		0	
插入记录	1	1	1

（3）虚表写入第二条记录，再次计算floor(rand(0)*2)，发现结果为1(此时为第三次计算)，此时结算结果为1，所以floor(rand(0)*2)不会被计算，直接count(*)加1，第二条记录写入完毕。（5）查询虚表，发现1的键值存在，所以floor(rand(0)*2)不会被计算第二次，直接count(*)加1，第二条记录查询完毕，结果如下:

操作	key	floor（rand（0）*2）	count（*）
取第一条记录		0	
插入记录	1	1	1
取第二条记录，不用插入	1	1	2

（4）虚表写入第三条记录，再次计算floor(rand(0)*2)，发现结果为0(此时为第4次计算)，计算结果为0，此时虚表中没有0的数据记录，则执行插入该数据，插入时会再次计算floor(rand(0)*2)（此时为第5次计算），计算结果为1。然而1这个主键已经存在于虚拟表中，而新计算的值也为1(主键键值必须唯一)，所以就产生了主键冲突的错误，也就是：Duplicate entry 的报错。

操作	key	floor（rand（0）*2）	count（*）
取第一条记录		0	
插入记录	1	1	1
取第二条记录，不用插入	1	1	2
取第三条记录		0	
插入记录	1?	1	

总结：

通过上述分析，在虚表中写入第三条记录是时，产生了报错。此时floor(rand(0)*2)一共被计算了5次，这也解释了为什么数据表中需要最少3条数据才会报错的原因。

另外，要注意加入随机数种子的问题，如果没加入随机数种子或者加入其他的数，那么floor(rand()*2)产生的序列是不可测的，这样可能会出现正常插入无法报错的情况。最重要的是前面几条记录查询后不能让虚表存在0,1键值，如果存在了，那无论多少条记录，也都没办法报错，因为floor(rand()*2)不会再被计算做为虚表的键值，这也就是为什么不加随机数种子有时候会报错，有时候不会报错的原因。

比如下面用1作为随机数种子，就不会产生报错：

```
mysql> select floor(rand(1)*2) x from users ;
+---+
| x |
+---+
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |
+---+
6 rows in set
```

```
mysql> select count(*),floor(rand(1)*2) x from users group by x;
+----------+---+
| count(*) | x |
+----------+---+
|        3 | 0 |
|        3 | 1 |
+----------+---+
2 rows in set
```