# mysql : Lock wait timeout exceeded; try restarting transaction - freeliver54 - 博客园

http://www.cnblogs.com/freeliver54/archive/2010/09/30/1839042.html

原因是你使用的InnoDB   表类型的时候,  
默认参数:innodb\_lock\_wait\_timeout设置锁等待的时间是50s,  
因为有的锁等待超过了这个时间,所以抱错.

你可以把这个时间加长,或者优化存储过程,事务避免过长时间的等待.

my.ini文件：  
#innodb\_lock\_wait\_timeout = 50  
-》  
innodb\_lock\_wait\_timeout = 500

重启mysql服务。

# 2@ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction - MySQL - 博客频道 - CSDN.NET

http://blog.csdn.net/mchdba/article/details/38313881

异常：Lock wait timeout exceeded; try restarting transaction  
  
解决办法：

执行select \* from information\_schema.innodb\_trx 之后找到了一个一直没有提交的只读事务， 找到对应的线程后，执行 kill thread id，再确认一直没有提交的只读事物被干掉了就OK了

**前言：朋友咨询我说执行简单的update语句失效，症状如下：**  
[**MySQL**](http://lib.csdn.net/base/mysql)> update order\_info  set province\_id=15  ,city\_id= 1667  where order\_from=10 and order\_out\_sn='1407261241xxxx';  
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction  
mysql>

**QQ远程过去，开始check  
1，查看**[**数据库**](http://lib.csdn.net/base/mysql)**的隔离级别：**

mysql> select @@tx\_isolation;  
+-----------------+  
| @@tx\_isolation  |  
+-----------------+  
| REPEATABLE-READ |  
+-----------------+  
1 row in set (0.00 sec)  
  
mysql>

**2，去查看先当前库的线程情况：**

mysql> show full processlist;

+----------+-----------------+-------------------+-----------------+-------------+---------+-------------------------+-----------------------+

| Id       | User            | Host              | db              | Command     | Time    | State                   | Info                  |

+----------+-----------------+-------------------+-----------------+-------------+---------+-------------------------+-----------------------+

|        1 | event\_scheduler | localhost         | NULL            | Daemon      | 9635385 | Waiting on empty queue  | NULL                  |

|  **9930577**| business\_web    | 192.168.1.21:45503 | business\_db     | Sleep       |     153 |                         | NULL                  |

|  9945825 | business\_web    | 192.168.1.25:49518 | business\_db     | Sleep       |      43 |                         | NULL                  |

|  9946322 | business\_web    | 192.168.1.23:44721 | business\_db     | Sleep       |     153 |                         | NULL                  |

|  9960167 | business\_web    | 192.168.3.28:2409  | business\_db     | Sleep       |      93 |                         | NULL                  |

|  9964484 | business\_web    | 192.168.1.21:24280 | business\_db     | Sleep       |       7 |                         | NULL                  |

|  9972499 | business\_web    | 192.168.3.28:35752 | business\_db     | Sleep       |      13 |                         | NULL                  |

| 10000117 | business\_web    | 192.168.3.28:9149  | business\_db     | Sleep       |       6 |                         | NULL                  |

| 10002523 | business\_web    | 192.168.3.29:42872 | business\_db     | Sleep       |       6 |                         | NULL                  |

| 10007545 | business\_web    | 192.168.1.21:51379 | business\_db     | Sleep       |     155 |                         | NULL                  |  
......

+----------+-----------------+-------------------+-----------------+-------------+---------+-------------------------+-----------------------+

没有看到正在执行的慢SQL记录线程，再去查看innodb的事务表INNODB\_TRX，看下里面是否有正在锁定的事务线程，看看ID是否在show full processlist里面的sleep线程中，如果是，就证明这个sleep的线程事务一直没有commit或者rollback而是卡住了，我们需要手动kill掉。

mysql> SELECT \* FROM information\_schema.INNODB\_TRX\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                    trx\_id: 20866

                 trx\_state: LOCK WAIT

               trx\_started: 2014-07-31 10:42:35

     trx\_requested\_lock\_id: 20866:617:3:3

          trx\_wait\_started: 2014-07-30 10:42:35

                trx\_weight: 2

       trx\_mysql\_thread\_id: **9930577**

                 trx\_query: delete from dltask where id=1

       trx\_operation\_state: starting index read

         trx\_tables\_in\_use: 1

         trx\_tables\_locked: 1

          trx\_lock\_structs: 2

     trx\_lock\_memory\_bytes: 376

           trx\_rows\_locked: 1

         trx\_rows\_modified: 0

   trx\_concurrency\_tickets: 0

       trx\_isolation\_level: READ COMMITTED

         trx\_unique\_checks: 1

    trx\_foreign\_key\_checks: 1

trx\_last\_foreign\_key\_error: NULL

 trx\_adaptive\_hash\_latched: 0

 trx\_adaptive\_hash\_timeout: 10000

          trx\_is\_read\_only: 0

trx\_autocommit\_non\_locking: 0

**3，看到有这条9930577的sql，kill掉，执行kill 9930577;**

mysql> kill 9930577;

Query OK, 0 rows affected (0.00 sec)

mysql>

然后再去查询INNODB\_TRX表，就没有阻塞的事务sleep线程存在了，如下所示：

mysql> SELECT \* FROM INNODB\_TRX\G;

Empty set (0.00 sec)

ERROR:

No query specified

mysql>

再去执行update语句，就能正常执行了，如下所示：

mysql> update order\_info  set province\_id=15  ,city\_id= 1667  where order\_from=10 and order\_out\_sn='1407261241xxxx';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1  Changed: 1  Warnings: 0

mysql>  
  
**4，总结分析**  
表数据量也不大，按照普通的情况来说，简单的update应该不会造成阻塞的，mysql都是autocommit，不会出现update卡住的情况，去查看下autocommit的值。  
mysql> select @@autocommit;  
+--------------+  
| @@autocommit |  
+--------------+  
|           **0**|  
+--------------+  
1 row in set (0.00 sec)  
  
  
mysql>  
  
看到亮闪闪的0，这个设置导致原来的update语句如果没有commit的话，你再重新执行update语句，就会等待锁定，当等待时间过长的时候，就会报ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction的错误。  
所以赶紧commit刚才执行的update语句，之后 set global autocommit=1;

来自：<http://blog.itpub.net/26230597/viewspace-1242948/>