<http://blog.csdn.net/fuzhongyu2/article/details/52729371>

# [事务控制和锁定语句](http://blog.csdn.net/fuzhongyu2/article/details/52729371)

IMG_256 分类：

mysql 不懂要快学啊（13） IMG_257

版权声明：本文为博主原创文章，未经博主允许不得转载。

**[MySQL](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)** 支持对MyIsam和memory存储引擎的表进行 表级锁定，对bdb存储引擎的表进行 行页级锁定，对InnoDB存储引擎的表进行 行级锁定。

默认情况下，表锁和行锁都是自动获得的，不需要额外的命令，但是在有的情况下，需要明确的进行锁表或者进行事务的控制，以确保整个事务的完整性，这样就需要使用事务控制和锁定语句来完成。

   一，lock table 和unlock table

    lock tables 可以锁定用于当前线程的表。如果表被其他线程锁定，则当前线程会等待，知道可以获取到所有锁定为止。

    unlock table可以释放当前线程获得的任何锁定。当前线程执行另一个lock tables 时，或当与服务器的连接被关闭时，所有由当前线程锁定的表被隐含的解锁。

下面我们演示一个例子：表t16获得read锁的情况，其他session更新该表记录 会等待锁，t16表释放锁以后，其他session可以进行更新操作。其中session1和session2表示两个同时打开的session，表格中的每一行表示同一时刻两个session的运行状况。

1. mysql> create table t16(
2. id **int**(11),
3. name varchar(63),
4. age **int**(3)
5. );
6. Query OK, 0 rows affected (0.31 sec)
8. mysql> desc t16;
9. +-------+-------------+------+-----+---------+-------+
10. | Field | Type        | Null | Key | Default | Extra |
11. +-------+-------------+------+-----+---------+-------+
12. | id    | **int**(11)     | YES  |     | NULL    |       |
13. | name  | varchar(63) | YES  |     | NULL    |       |
14. | age   | **int**(3)      | YES  |     | NULL    |       |
15. +-------+-------------+------+-----+---------+-------+
16. 3 rows in set (0.00 sec)
18. mysql> insert into  t16(id,name,age)values(1,'fzy1',1),(1,'fzy1',1),(3,'fzy3',3);
19. Query OK, 3 row affected (0.13 sec)

|  |  |  |
| --- | --- | --- |
| 获得表锁和释放表锁事例 | | |
| **session\_1** | **session\_2** | |
| 获得表t16的read锁定   1. mysql> lock  table  t16  read; 2. Query OK, 0 rows affected (0.00 sec) |  | |
| 当session可以查询该表记录  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_262](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_263](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=1; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    1 | fzy1 |    1 | 6. +------+------+------+ 7. 1 row in set (0.00 sec) | 其他session也可以查询该表的记录  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_264](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_265](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=1; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    1 | fzy1 |    1 | 6. +------+------+------+ 7. 1 row in set (0.00 sec) | |
|  | | 其他session更新锁定的表会等待获得锁  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_266](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_267](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)  Database  changed  mysql> update t16 set age=3 where id=1;    （一直处于光标闪烁，即等待状态） |
| 释放锁  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_268](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_269](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> unlock tables; 2. Query OK, 0 rows affected (0.00 sec) | | 等待 |
|  | | session 获得锁，更新操作完成  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_270](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_271](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. Query OK, 1 row affected (1 min 38.38 sec) 2. Rows matched: 1  Changed: 1  Warnings: 0 |

二，事务控制

   mysql 通过set autocommit , start transaction , commit , rollback等语句支持本地事务。

默认情况下，mysql是自动提交(autocommit)的，如果需要明确的commit和rollback来提交和回滚事务，那么就需要通过明确的事务控制命令来开始事务，这是和**[Oracle](http://lib.csdn.net/base/oracle" \o "Oracle知识库" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)**的事务管理明显不同的地方。

    （1）start transaction 或 begin语句可以开始一项新的事务

    （2）commit 和rollback用来提交或者回滚事务。

     （3）chain和release字句分别用来定义在事务提交或回滚之后的操作。

chain会立即启动一个新事务，并且和刚才的事务具有相同的隔离级别。release则会断开和客户端的连接。

    （4）set autocommit可以修改当前连接的提交方式，如果设置了set autocommit=0，则设置之后的所有事务读需要通过明确的命令进行提交或者回滚。

   如果只是对某些语句需要进行事务控制，则使用start transaction语句开始一个事务比较方便，这样事务结束之后可以自动回到自动提交的方式。

如果希望所有的事务都不是自动提交的，那么通过修改autocommit来控制事务比较方便。

使用start transcation开始事务在提交后 自动回到 自动提交的方式。如果在提交的时候使用commit and chain那么在提交后立即开始一个新的事务。

|  |  |
| --- | --- |
| start transation和commit and chain事例 | |
| **session\_1** | **session\_2** |
| 从表t16中查询id=4，结果为空  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_272](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_273](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=4; 2. Empty set (0.00 sec) | 从表t16中查询id=4，结果为空  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_274](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_275](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=4; 2. Empty set (0.00 sec) |
| 用start transaction命令启动一个事务，往表t16中插入一条记录，没有commit  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_276](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_277](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> start transaction; 2. Query OK, 0 rows affected (0.00 sec) 4. mysql> insert into t16(id,name,age)values(4,'fzy4',4); 5. Query OK, 1 row affected (0.01 sec) |  |
| 查询id=4,有结果  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_278](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_279](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=4; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    4 | fzy4 |    4 | 6. +------+------+------+ 7. 1 row in set (0.00 sec) | 查询表t16 id=4,结果仍为空  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_280](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_281](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=4; 2. Empty set (0.00 sec) |
| 执行提交事务  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_282](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_283](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> commit; 2. Query OK, 0 rows affected (0.04 sec) |  |
|  | 再次查询  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_284](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_285](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=4; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    4 | fzy4 |    4 | 6. +------+------+------+ 7. 1 row in set (0.00 sec) |
| 这个事务是按照 自动提交执行的，start transcation开始事务在提交后 自动回到 自动提交的方式。   1. mysql> insert into t16(id,name,age)values(5,'fzy5',5); 2. Query OK, 1 row affected (0.04 sec) |  |
|  | 查询插入的记录  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_288](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_289](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=5; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    5 | fzy5 |    5 | 6. +------+------+------+ 7. 1 row in set (0.01 sec) |
| 重新用start transaction启动一个事务  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_290](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_291](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> start transaction; 2. Query OK, 0 rows affected (0.00 sec) 4. ##插入语句 5. mysql> insert into t16(id,name,age)values(6,'fzy6',6); 6. Query OK, 1 row affected (0.00 sec) 8. ##提交并立即开启一个新事务（和刚才具有相同的隔离级别） 9. mysql> commit and chain; 10. Query OK, 0 rows affected (0.04 sec) 12. 插入数据 13. mysql> insert into t16(id,name,age)values(7,'fzy7',7); 14. Query OK, 1 row affected (0.00 sec) |  |
|  | 查询   1. mysql> select \* from t16; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    1 | fzy1 |    3 | 6. |    2 | fzy2 |    2 | 7. |    3 | fzy3 |    3 | 8. |    4 | fzy4 |    4 | 9. |    5 | fzy5 |    5 | 10. |    6 | fzy6 |    6 | 11. +------+------+------+ 12. 6 rows in set (0.00 sec)   发现id=7并没有这条记录 |
| 再提交事务  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_294](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_295](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> commit; 2. Query OK, 0 rows affected (0.05 sec) |  |
|  | 查询   1. mysql> select \* from t16; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    1 | fzy1 |    3 | 6. |    2 | fzy2 |    2 | 7. |    3 | fzy3 |    3 | 8. |    4 | fzy4 |    4 | 9. |    5 | fzy5 |    5 | 10. |    6 | fzy6 |    6 | 11. |    7 | fzy7 |    7 | 12. +------+------+------+ 13. 7 rows in set (0.01 sec)   id=7这条记录已经插入 |

如果在锁表期间，用start transaction命令开始一个新事务，会造成一个隐含的unlock tables被执行

|  |  |  |  |
| --- | --- | --- | --- |
| start transaction导致的unlock tables | | | |
| **session\_1** | | **session\_2** | |
| 从表t16中查询id=9的记录，结果为空  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_298](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_299](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=9; 2. Empty set (0.00 sec) | | 从表t16中查询id=9的记录，结果为空  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_300](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_301](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=9; 2. Empty set (0.00 sec) | |
| 对表t16加写锁  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_302](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_303](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> lock table t16 write; 2. Query OK, 0 rows affected (0.00 sec) | |  | |
|  | | | 对表t16的读操作被阻塞  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_304](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_305](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16; 2. 等待 |
| 插入一条记录  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_306](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_307](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> insert into t16(id,name,age)values(9,'fzy9',9); 2. Query OK, 1 row affected (0.06 sec) | 等待 | | |
| 回滚刚才记录  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_308](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_309](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> rollback; 2. Query OK, 0 rows affected (0.00 sec) | 等待 | | |
| 用start transaction命令开始一个事务  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_310](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)[IMG_311](https://code.csdn.net/snippets/1912279/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> start transaction; 2. Query OK, 0 rows affected (0.00 sec) | 等待 | | |
|  | session1开始一个事务的时候，表锁被释放，  可以查询   1. mysql> select \* from t16; 2. +------+------+------+ 3. | id   | name | age  | 4. +------+------+------+ 5. |    1 | fzy1 |    3 | 6. |    2 | fzy2 |    2 | 7. |    3 | fzy3 |    3 | 8. |    4 | fzy4 |    4 | 9. |    5 | fzy5 |    5 | 10. |    6 | fzy6 |    6 | 11. |    7 | fzy7 |    7 | 12. |    8 | fzy8 |    8 | 13. |    9 | fzy9 |    9 | 14. +------+------+------+ 15. 9 rows in set (0.00 sec)   对lock方式加的表锁，不能通过rollback进行回滚 | | |

   因此，在同一个事务中，最好不使用不同的存储引擎，否则rollback时需要对非事务类型的表进行特别的处理。因为，commit ,rollback 只能对事务类型的表进行提交和回滚。

   通常情况下，只对提交的事务记录到二进制的日志中，但是如果一个事务中包含非事务类型的表，那么回滚操作也会被记录到二进制日志中，以确保非事务类型表的更新可以被复制到从**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)**(slave)中。

  注：所有的ddl语句是不能回滚的，并且部分ddl语句会造成隐式的提交。

DDL语句用语定义和管理数据库中的对象，如Create,Alter和Drop.

     在事务中可以通过定义savepoint，指定回滚事务部分，但是不能指定提交事务的一个部分。

对于复杂的应用，可以定义多个不同的savepoint,满足不同的条件时，回滚不同的savepoint。需要注意的是，如果定义了相同名字的savepoint，则后面定义的savepoint会覆盖之前的定义。

对于不需要使用的savepoint，可以通过release savepoint命令删除。

    下面我们模拟回滚事务：

|  |  |  |
| --- | --- | --- |
| 模拟回滚事务 | | |
| **session\_1** | | **session\_2** |
| 从表t16中查询id=10,为空  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_314](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=10; 2. Empty set (0.00 sec) | | 从表t16中查询id=10，为空  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_316](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=10; 2. Empty set (0.00 sec) |
| 启动一个事务，往表里插入数据id=10  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_318](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> start transaction; 2. Query OK, 0 rows affected (0.00 sec) 4. mysql> insert into t16(id,name,age)values(10,'fzy10',10); 5. Query OK, 1 row affected (0.00 sec) |  | |
| 可以查到插入的数据  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_320](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=10; 2. +------+-------+------+ 3. | id   | name  | age  | 4. +------+-------+------+ 5. |   10 | fzy10 |   10 | 6. +------+-------+------+ 7. 1 row in set (0.00 sec) | | 不能查到插入的数据  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_322](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=10; 2. Empty set (0.00 sec) |
| 定义savepoint ,名称为test\_savepoint1，接着再插入一条数据  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_324](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> savepoint test\_savepoint; 2. Query OK, 0 rows affected (0.00 sec) 4. mysql> insert into t16(id,name,age)values(11,'fzy11',11); 5. Query OK, 1 row affected (0.00 sec) | |  |
| 可以查询到插入的2条记录  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_326](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> select \* from t16 where id=11; 2. +------+-------+------+ 3. | id   | name  | age  | 4. +------+-------+------+ 5. |   11 | fzy11 |   11 | 6. +------+-------+------+ 7. 1 row in set (0.00 sec | | 仍不能查询到插入的2条记录  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_328](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)  mysql> select \* from t16 where id=10 or id=11;  Empty set (0.00 sec) |
| 回滚刚才定义的savepoint  **[java]** [view plain](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "view plain) [copy](http://blog.csdn.net/fuzhongyu2/article/details/52729371" \o "copy) [IMG_330](https://code.csdn.net/snippets/1912279" \o "在CODE上查看代码片" \t "http://blog.csdn.net/fuzhongyu2/article/details/_blank)   1. mysql> rollback to savepoint test\_savepoint; 2. Query OK, 0 rows affected (0.00 sec) | |  |
| 只能从表中查询到第一条记录   1. mysql> select \* from t16 where id=11 or id=10; 2. +------+-------+------+ 3. | id   | name  | age  | 4. +------+-------+------+ 5. |   10 | fzy10 |   10 | 6. +------+-------+------+ 7. 1 row in set (0.00 sec) | | 仍不能查询到刚插入的记录  mysql> select \* from t16 where id=10 or id=11;  Empty set (0.00 sec) |
| 提交   1. mysql> commit; 2. Query OK, 0 rows affected (0.05 sec) | |  |
| 只能查到一条记录   1. mysql> select \* from t16 where id=11 or id=10; 2. +------+-------+------+ 3. | id   | name  | age  | 4. +------+-------+------+ 5. |   10 | fzy10 |   10 | 6. +------+-------+------+ 7. 1 row in set (0.00 sec) | | 能查到一条记录  mysql> select \* from t16 where id=11 or id=10;   1. +------+-------+------+ 2. | id   | name  | age  | 3. +------+-------+------+ 4. |   10 | fzy10 |   10 | 5. +------+-------+------+ 6. 1 row in set (0.00 sec) |

 三，分布式事务（xa）的使用

       当前分布式事务只支持innodb存储引擎，一个分布式事务会涉及多个行动，这些行动本身是事务性的，所有行动都必须一起成功完成，或者一起被回滚。

  1，分布式事务原理

       在mysql中，使用分布式事务的应用程序涉及一个或多个资源管理器和一个事务管理器。