MySQL_REPEATABLE-READ事务隔离级别 && 幻读

关于mysql命令行中事务控制的语句见该文章

http://my.oschina.net/xinxingegeya/blog/296459

关于MVCC多版本控制

http://my.oschina.net/xinxingegeya/blog/208821

表结构

```
create table t1(
    a int primary key,
    b int not null
)
```

REPEATABLE-READ可重复读（一）

这里打开两个mysql的命令行窗口，窗口A，即session1，窗口B，即session2。

session1

```
mysql> begin ;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
+----+------+
4 rows in set (0.00 sec)
```

以上sql只是显示的开启了事务，执行了sql查询。下面看session2的操作。

要注意这里的select操作是一般的快照读。根据MVCC多版本控制规则读取的数据行。

session2

```
mysql> begin ;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into t1 values (55, 3000);
Query OK, 1 row affected (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.03 sec)

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
| 55 | 3000 |
+----+------+
5 rows in set (0.00 sec)
```

session2插入了一条数据，并显式的提交了事务。

session1

此时返回session1进行以下操作

```
mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
+----+------+
4 rows in set (0.00 sec)
```

此时，虽然在session2中插入了一条数据，并且提交了事务，但***在session1中的查询和session1的上次查询还是同一个结果，这就是重复读。（也可以说是根据MVCC规则读取的数据行）。***如果是在"READ-COMMITTED"级别下是可以读到a=55这条记录的（因为session2在刚才已经提交了事务）。

REPEATABLE-READ可重复读（二）

session1

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from t1 where a = 56;
+----+------+
| a  | b    |
+----+------+
| 56 | 7000 |
+----+------+
1 row in set (0.00 sec)
```

session1，开启了一个事务，查询a = 56 的记录。

session2

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> update t1 set b = 8000 where a = 56;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from t1 where a = 56;
+----+------+
| a  | b    |
+----+------+
| 56 | 8000 |
+----+------+
1 row in set (0.00 sec)

mysql> commit
    -> ;
Query OK, 0 rows affected (0.04 sec)
```

session2开启了事务，更新a = 56 的记录，同时查询a = 56 的记录，可以看到在同一事务内重复读的效果。

session1

```
mysql> select * from t1 where a = 56;
+----+------+
| a  | b    |
+----+------+
| 56 | 7000 |
+----+------+
1 row in set (0.00 sec)
```

和上次查询结果一致，验证了重复读。还是要注意这里的select操作只是一般的快照读。其实不管session2 做什么操作，这里的快照读都是重复读的。

此时，如果***session1***提交该事务，重新开启事务，查询能查到session2中修改的结果

```
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from t1 where a = 56;
+----+------+
| a  | b    |
+----+------+
| 56 | 8000 |
+----+------+
1 row in set (0.00 sec)
```

注：以上的重复读，虽然在当前事务中真的是重复读的现象，但到底来说是通过MVCC多版本控制实现的可重复读。

注：以上的重复读，虽然在当前事务中真的是重复读的现象，但到底来说是通过MVCC多版本控制实现的可重复读。

REPEATABLE-READ与幻读

session1

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
| 55 | 4000 |
| 56 | 8000 |
+----+------+
6 rows in set (0.00 sec)
```

开启事务，select操作为快照读。

session2

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
| 55 | 4000 |
| 56 | 8000 |
+----+------+
6 rows in set (0.00 sec)

mysql> insert into t1 values (57, 1000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
| 55 | 4000 |
| 56 | 8000 |
| 57 | 1000 |
+----+------+
7 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.11 sec)
```

在session2中做了一系列的操作，插入insert，这里其实是当前读（写入）。然后提交事务。

session1

```
mysql> update t1 set b = b+1000;
Query OK, 7 rows affected (0.00 sec)
Rows matched: 7  Changed: 7  Warnings: 0
```

session1做更新操作，这里更新成功。如果session2 插入记录后，没有提交事务，这里更新是要阻塞的，因为session2 插入记录持有那条记录的X锁。

session1整个会话的sql

```
mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 4000 |
| 52 | 4000 |
| 53 | 4000 |
| 54 | 4000 |
| 55 | 5000 |
| 56 | 9000 |
| 57 | 2000 |
+----+------+
7 rows in set (0.00 sec)
```

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
| 55 | 4000 |
| 56 | 8000 |
+----+------+
6 rows in set (0.00 sec)

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 3000 |
| 52 | 3000 |
| 53 | 3000 |
| 54 | 3000 |
| 55 | 4000 |
| 56 | 8000 |
+----+------+
6 rows in set (0.00 sec)

mysql> update t1 set b = b+1000;
Query OK, 7 rows affected (0.00 sec)
Rows matched: 7  Changed: 7  Warnings: 0

mysql> select * from t1;
+----+------+
| a  | b    |
+----+------+
| 51 | 4000 |
| 52 | 4000 |
| 53 | 4000 |
| 54 | 4000 |
| 55 | 5000 |
| 56 | 9000 |
| 57 | 2000 |
+----+------+
7 rows in set (0.00 sec)

mysql>
```

可以看到多出了一行，这算是幻读吗？其实不是。这是根据MVCC的select规则读出来的数据

请详见http://mysqlchina.net/xinxingegeya/blog/505675