实验12 数据库并发事务实验

学号	**19335162**	姓名	潘思晗

一、实验目的

熟悉数据库并发事务的相关操作实验。

二、实验内容及结果

1.锁的获得与释放

编写事务程序，用于更新**book**表中的**SQL Server 2012**宝典的信息，观察其事务过程中锁的获得与释放情况，以及锁定资源的类型。

更新事务的SQL语句如下（便于观察锁，没有提交事务或回滚）：

```
begin transaction
    update book set book_price = book_price + 4 where book_id = 'b0001'
```

使用sp_lock存储过程查看锁的获得与释放情况：

```
Exec sp_lock
GO
```

更新之前，查看锁如下：

结果 消息

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	67	6	0	0	DB		S	GRANT
2	67	1	1787153412	0	TAB		IS	GRANT
3	67	32767	-571204656	0	TAB		Sch-S	GRANT
4	69	4	0	0	DB		S	GRANT

更新之后，查看锁的状态如下：

结果 消息

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	63	6	0	0	DB		S	GRANT
2	63	1	1787153412	0	TAB		IS	GRANT
3	63	32767	-571204656	0	TAB		Sch-S	GRANT
4	67	6	0	0	DB		S	GRANT
5	67	6	1013578649	1	PAG	1:440	IX	GRANT
6	67	6	1013578649	1	KEY	(4e6f9aad940e)	X	GRANT
7	67	6	1013578649	0	TAB		IX	GRANT
8	69	4	0	0	DB		S	GRANT

这就是更新事务执行过程中获得锁的信息，其中spid是请求锁进程的数据库会话引擎ID，此处为63；dbid是保留锁的数据库的ID；ObjId是持有锁的对象ID；IndId是持有锁的表索引ID，0表示数据表或数据库的锁；Type是资源持有锁的锁粒度类型，DB表示数据库，TAB表示包括所有数据和索引在内的整个表。

2.隔离级别

编写事务程序，对**book**表进行实验，设置相应的隔离级别，模拟实现读脏数据、不可重复读以及可重复读。

（**1**）设置未提交读隔离级别，实现脏读和不可重复读

建立两个连接，连接1的SQL语句如下，在连接中更新数据，将价格的89元修改为85元，延时10秒后回滚数据：

```sql
begin transaction
    update book set book_price = 85 where book_id = 'b0001'
    waitfor delay '00:00:10'
    select * from book where book_id = 'b0001'
rollback transaction
```

连接2的SQL语句如下所示，模拟实现脏读和不可重复读：

```sql
set transaction isolation level READ UNCOMMITTED
    select * from book where book_id = 'b0001'
    if @@ROWCOUNT <> 0
        begin
            waitfor delay '00:00:10'
            select * from book where book_id = 'b0001'
        end
```

输入上述语句并执行，执行之后，可以看到连接2的执行结果如下：

结果 消息

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	85.00

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	89.00

从结果可以看出，连接2第一次读到的book_price数据是事务1没有提交的数据，当连接2第二次读取数据时，连接1回滚了，所以读到的数据和第一次不一致。此处第一次读取发生了数据脏读，第二次发生了不可重复读。

发生脏读的原因是连接1执行过程中没有和连接2相互隔离，导致连接2重复读取了连接1没有提交的数据。

（**2**）设置提交读隔离级别，避免脏读，实现不可重复读

建立两个连接，连接1的SQL语句如下，在连接中重复两次相同的查询：

```sql
set transaction isolation level READ COMMITTED
    begin transaction
    select * from book where book_id = 'b0001'
    if @@ROWCOUNT <> 0
        begin
            waitfor delay '00:00:10'
            select * from book where book_id = 'b0001'
        end
rollback transaction
```

连接2的SQL语句如下所示，更新记录，将价格改回89元：

```sql
set transaction isolation level READ COMMITTED
    update book set book_price = 89 where book_id = 'b0001'
```

输入上述语句并执行，执行之后，可以看到连接1的执行结果如下：

结果 消息

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	85.00

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	89.00

从结果可以看出，读出同一个数据项的值时是不同的，这就是不可重复读。

（3）设置可重复读隔离级别，避免脏读和不可重复读

建立两个连接，连接1的SQL语句如下，隔离级别为可重复读：

```sql
set transaction isolation level REPEATABLE READ
    begin transaction
    select * from book where book_id = 'b0001'
    if @@ROWCOUNT <> 0
        begin
            waitfor delay '00:00:10'
            select * from book where book_id = 'b0001'
        end
rollback transaction
```

连接2的SQL语句如下所示，更新记录，将价格设置为90元：

```
set transaction isolation level REPEATABLE READ
    update book set book_price = 90 where book_id = 'b0001'
```

输入上述语句并执行，执行之后，可以看到连接1的执行结果如下：

结果 消息

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	89.00

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	89.00

从结果可以看出，设置隔离级别为repeatable read时，同一事务内部对同一数据的读取，其结果是一致的。

3.死锁

编写事务程序，对**book**表进行实验，设计实验制造事务之间的死锁。

建立两个连接，同时执行如下SQL语句：

```
set transaction isolation level REPEATABLE READ
waitfor delay '00:00:05'
    begin transaction
        select * from book where book_id = 'b0001'
        waitfor delay '00:00:10'
        update book set book_price = 80 where book_id = 'b0001'
    commit transaction
select * from book where book_id = 'b0001'
```

执行之后，可以看到其中一个连接的执行结果如下：

结果 消息

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	90.00

	book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
1	b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	18	80.00

而另一个连接的结果如下：

结果 消息

```
(1 行受影响)
消息 1205，级别 13，状态 51，第 6 行
事务(进程 ID 67)与另一个进程被死锁在 锁 资源上，并且已被选作死锁牺牲品。请重新运行该事务。
```

出现死锁的原因是两个连接都是通过设置共享锁对同一数据进行查询，并接着尝试转换为更新锁，进而到排他锁以完成更新操作，但是设置事务的隔离级别为可重复读，在事务完成之前，两个连接不可能释放共享锁而永远无法更新，因而导致发生死锁。

三、实验总结

本次实验主要是了解和熟悉数据库并发事务的相关操作实验。通过本次实验，我学习了锁、隔离级别、死锁的相关知识，很有收获。