

Lecture 15, Fall 2017/2018

# 数据库系统实验

Yubao Liu (刘玉葆)

School of Data and Computer Science

Sun Yat-sen University

- 本节课提纲

- 实验目的
- 实验内容
- 实验示例
- 练习

- 实验目的

学会识别锁冲突，学会检查和处理死锁。

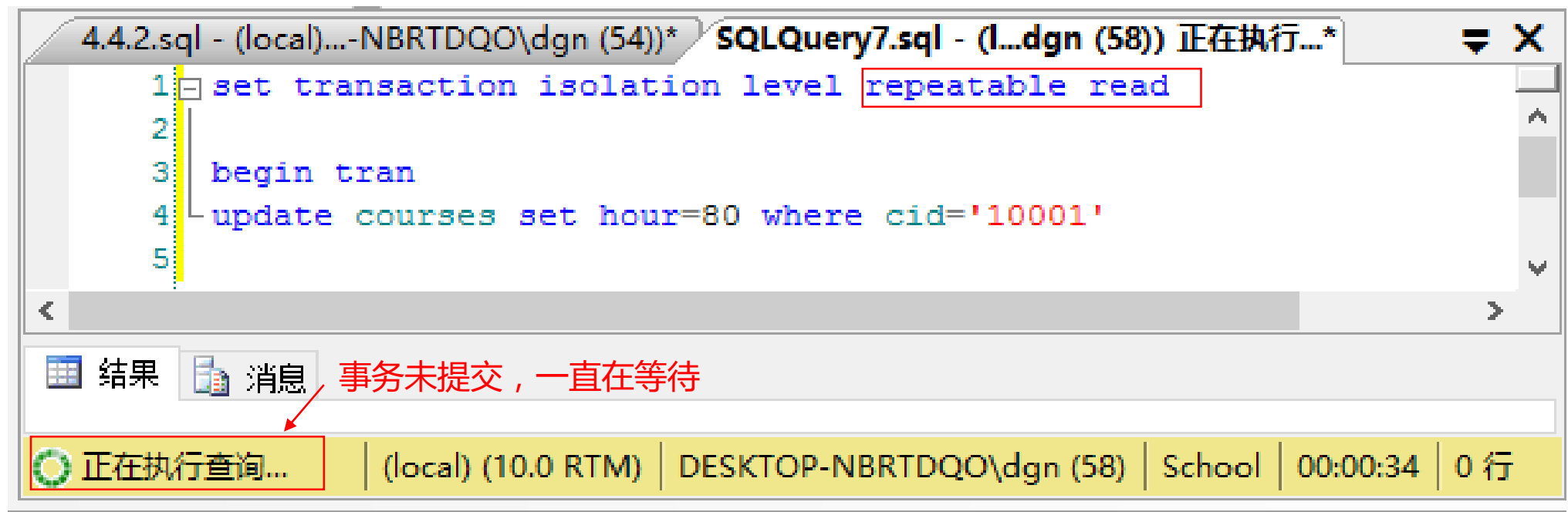
- 实验内容

1. 设计实验造成事务对资源的争夺，分析原因，讨论解决锁争夺的办法。
2. 设计实验制造事务之间的死锁，分析造成死锁的原因。

# • 实验示例

1.演示事务对资源的争夺。

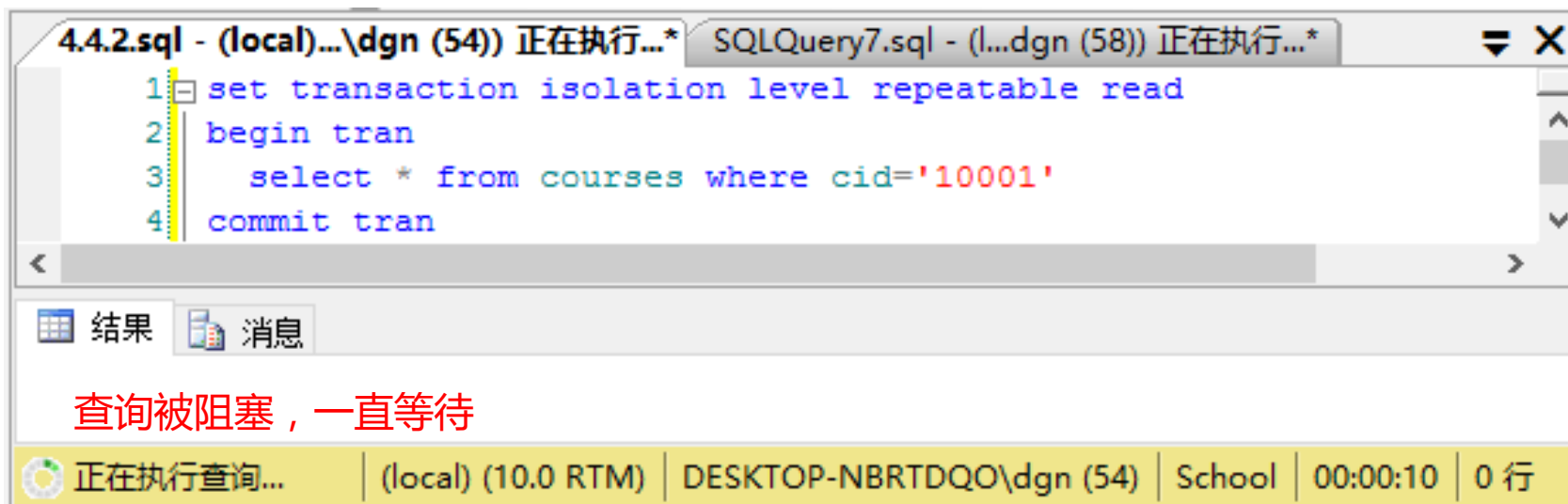
Step1：建立一个连接，更新courses表中的database的课时（为了制造锁争夺，更新事务没有提交）。



注：设置repeatable read隔离级别，如果某行在被读取之后由事务进行了修改，则该事务会获取一个用于保护该行的排他锁，并且该排他锁在事务完成之前将一直保持（即不允许其他事务对该行进行读取或者修改）。

# 实验示例

Step2：建立第二个连接，执行查询事务，发现被连接1阻塞。



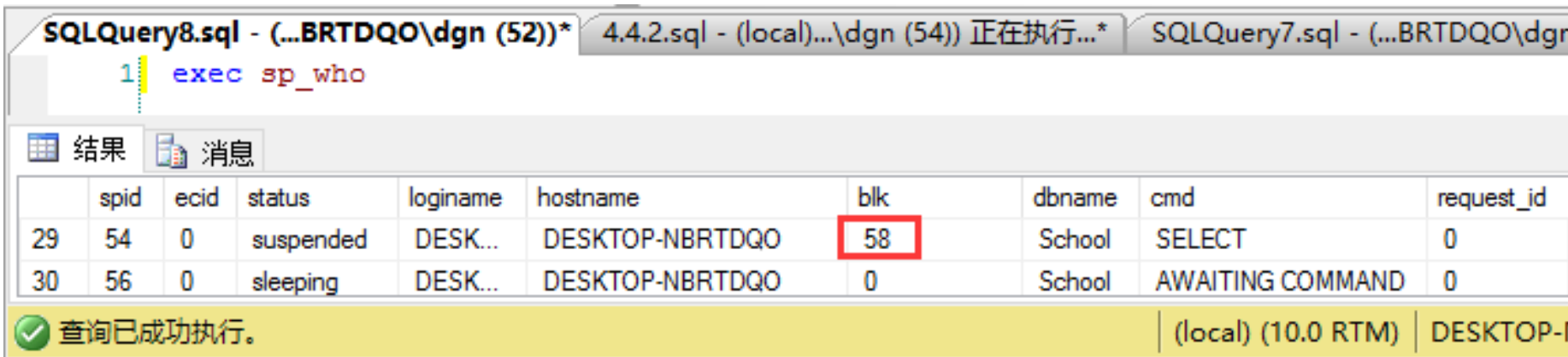
The screenshot shows a SQL query window titled "4.4.2.sql - (local)...\dgn (54)) 正在执行...". The query contains the following SQL code:

```
1 set transaction isolation level repeatable read
2 begin tran
3     select * from courses where cid='10001'
4 commit tran
```

Below the query editor, the "消息" (Messages) tab is selected, displaying the message: "查询被阻塞，一直等待" (Query is blocked, waiting indefinitely).

The status bar at the bottom indicates: "正在执行查询..." (Executing query...), "(local) (10.0 RTM)", "DESKTOP-NBRTDQO\dgn (54)", "School", "00:00:10", and "0 行" (0 rows).

注：验证进程是否受阻，可通过“exec sp\_who”，检查blk列非0值（即阻塞），如下图说明进程54被进程58阻塞。



The screenshot shows a SQL query window titled "SQLQuery8.sql - (...BRTDQO\dgn (52))\*". The query contains the command: "exec sp\_who".

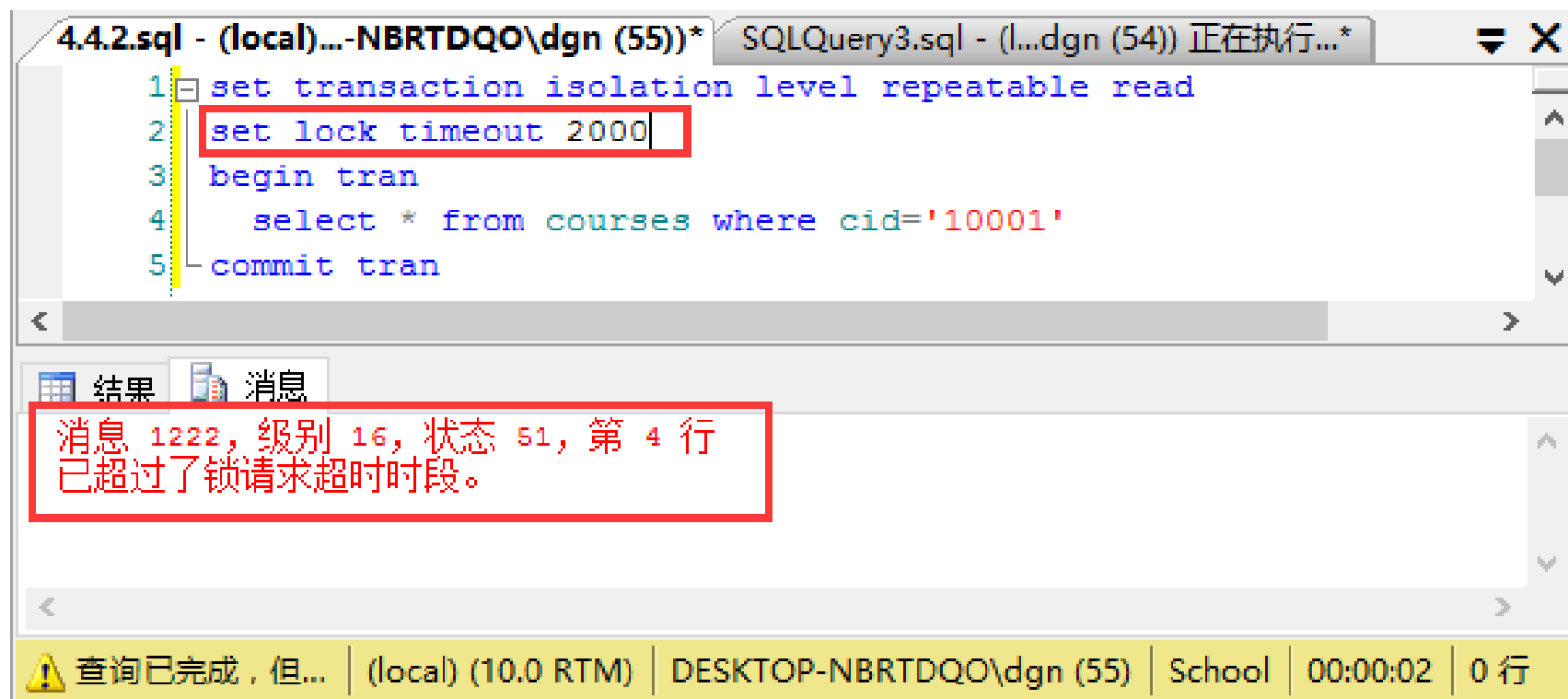
Below the query editor, the "结果" (Results) tab is selected, displaying a table with the following data:

	spid	ecid	status	loginame	hostname	blk	dbname	cmd	request_id
29	54	0	suspended	DESK...	DESKTOP-NBRTDQO	58	School	SELECT	0
30	56	0	sleeping	DESK...	DESKTOP-NBRTDQO	0	School	AWAITING COMMAND	0

The status bar at the bottom indicates: "查询已成功执行。" (Query executed successfully.), "(local) (10.0 RTM)", and "DESKTOP-I".

## • 实验示例

为了解决“永久等待”，可通过lock\_timeout设置锁定超时时间间隔。超时后，锁定管理器将自动解除锁的争夺。



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL code:

```
1 set transaction isolation level repeatable read
2 set lock timeout 2000
3 begin tran
4 select * from courses where cid='10001'
5 commit tran
```

The second line, `set lock timeout 2000`, is highlighted with a red box. The bottom pane shows the "Messages" tab with the following error message:

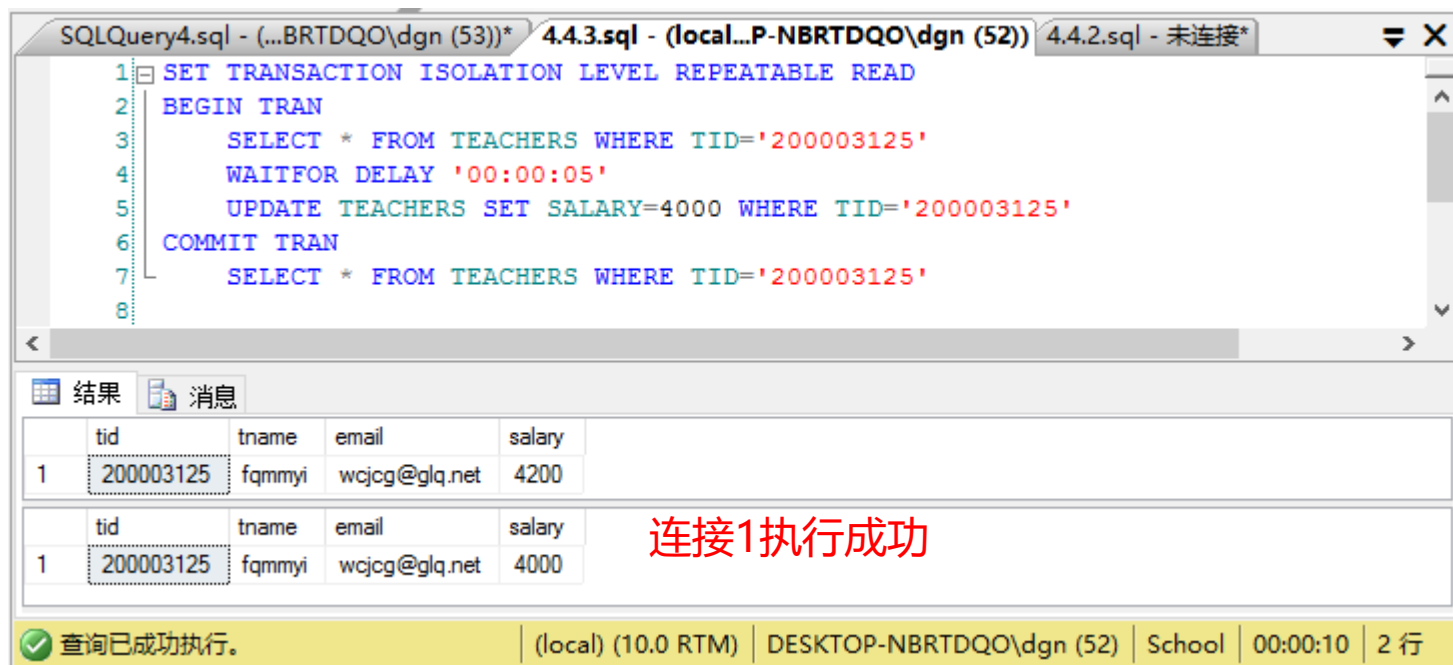
消息 1222, 级别 16, 状态 51, 第 4 行  
已超过了锁请求超时时段。

The status bar at the bottom indicates: 查询已完成, 但... | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (55) | School | 00:00:02 | 0 行

# 实验示例

## 2.演示事务间的死锁。

**打开两个连接，同时执行下面的代码。**可以发现有一个连接可以查询，另一个连接由于死锁，直接停掉了当前程序工作，并回滚之前的事务。



The screenshot shows a SQL query window with the following content:

```
1 SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
2 BEGIN TRAN
3     SELECT * FROM TEACHERS WHERE TID='200003125'
4     WAITFOR DELAY '00:00:05'
5     UPDATE TEACHERS SET SALARY=4000 WHERE TID='200003125'
6 COMMIT TRAN
7     SELECT * FROM TEACHERS WHERE TID='200003125'
8
```

Below the query, the results are displayed in a table:

	tid	tname	email	salary
1	200003125	fqmyi	wcjcg@glq.net	4200

Below the first table, the results are displayed in a second table:

	tid	tname	email	salary
1	200003125	fqmyi	wcjcg@glq.net	4000

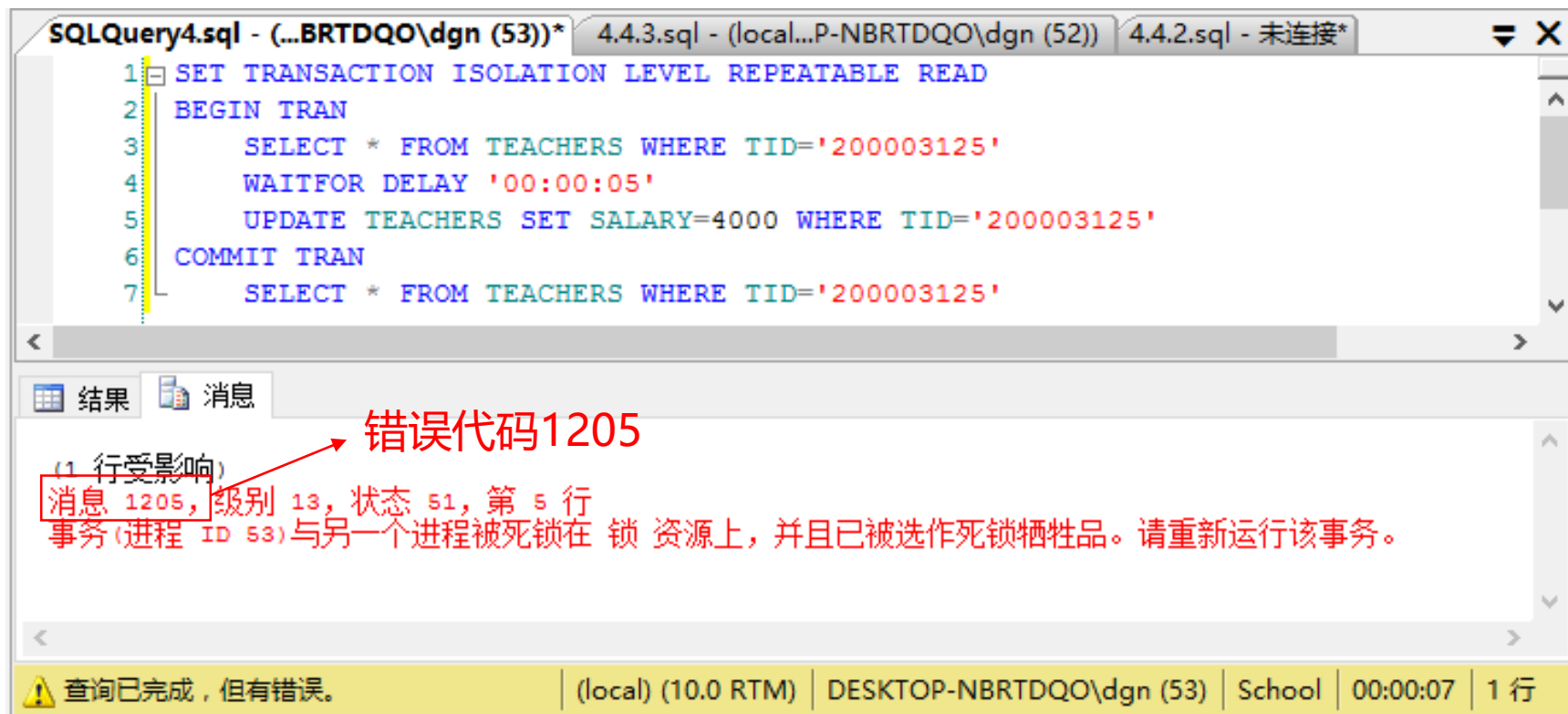
Red text "连接1执行成功" (Connection 1 execution successful) is overlaid on the second table.

At the bottom, a status bar shows: 查询已成功执行。 (local) (10.0 RTM) DESKTOP-NBRTDQO\dgn (52) School 00:00:10 2 行



# 实验示例

连接2由于死锁，报错：



出现死锁的原因：因为两个连接都通过设置共享锁（shared lock）对同一数据进行查询，并尝试转换为更新锁（update lock），进而到排它锁（exclusive）以完成更新操作。但隔离级别为“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，因而导致死锁

- 实验示例

注：

- 1.当系统发生锁争夺时，如果有事务超时，SQL Sever向用户返回错误号1222，当发生死锁时，如果有牺牲事务，SQL Sever向用户返回错误号1205，在应用时，需要在应用程序中处理锁争夺与死锁，通过在错误处理器中捕获消息1222或1205，然后让应用自动重新提交事务。
- 2.为了避免死锁，存取资源顺序最好相同。如连接A先存取甲数据库对象，再存取乙数据库对象，如果连接B的存取顺序刚好相反，则有可能发生死锁。

- 练习

以下练习均在school数据库中students表上进行。

- 1.在students表上演示锁争夺，通过sp\_who查看阻塞的进程。通过设置lock\_timeout解除锁争夺。
- 2.在students表上演示死锁。
- 3.讨论如何避免死锁以及死锁的处理方法。

死锁与回滚

尽量使用s锁

注意phase的分段