

多粒度封锁



引言

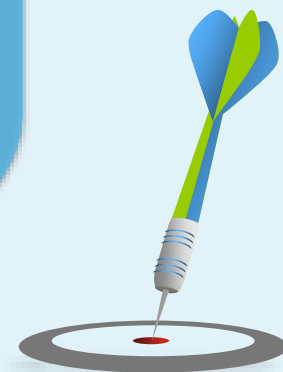
数据库对象

- **逻辑单元**
 - 关系、元组、属性值、属性值集合、索引项、整个索引、整个数据库
- **物理单元**
 - 磁盘数据块（页）、磁盘索引块（页）、物理记录



讲授内容

- 1 封锁粒度
- 2 多粒度封锁模式





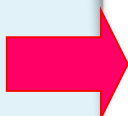
封锁粒度

- 封锁粒度：封锁对象单元的大小
 - 细粒度锁：元组锁（行级锁）
 - 粗粒度锁：关系锁（表级锁）
- 细粒度锁比粗粒度锁使事务具有更好的并发度



封锁粒度

关系锁



元组锁



元组锁



A\SQLEXPRESS.tr...e - dbo.Accounts X

	acctNo	balance
	A	1500
	B	0
*	NULL	NULL





封锁粒度

- 封锁粒度的选择需同时考虑管理锁的开销和事务并发度两个因素。
- DBMS实现了**多粒度封锁**功能，来满足不同应用需求。
 - 处理大量元组的事务可以选择关系为封锁粒度
 - 只处理少量元组的用户事务可选择元组为封锁粒度
 - 事务可根据操作需要来选择封锁粒度



多粒度封锁模式

多粒度封锁模式实例

- NOLOCK: 读数据前不加锁（仅用于SELECT语句）。
- TABLOCK: 在表上加共享锁，在读完数据后立即释放锁。
- HOLDLOCK: 保持锁，用于TABLOCK后，将共享锁保留到事务完成。
- UPDLOCK: 在满足条件的元组上加更新锁。
- TABLOCKX: 在表上加排他锁。



多粒度封锁模式

多粒度封锁模式实例

事务T2申请的锁

事务T1获得的锁

T2		事务T2申请的锁				
T1		NOLOCK	TABLOCK	HOLDLOCK	UPDLOCK	TABLOCKX
NOLOCK		Y	Y	Y	Y	Y
TABLOCK		Y	Y	Y	N	N
HOLDLOCK		Y	Y	Y	N	N
UPDLOCK		Y	N	N	Y(不同元组) N(相同元组)	N
TABLOCKX		Y	N	N	N	N



多粒度封锁模式

多粒度封锁模式实例

- 封锁操作是在SELECT、INSERT、UPDATE和DELETE等语句中添加WITH子句来完成的。

```
SELECT SNO, GRADE  
FROM SC WITH(TABLOCK)  
WHERE SNO='S01'
```



多粒度封锁模式

- ▶ 基于创建的学生选课数据库，在DBMS上并发执行两个事务，分析封锁的结果，并观察多粒度封锁对事务的并发性和数据一致性的影响。

S (SNO, SN, SD, SB, SEX)

C (CNO, CN, PC)

SC (SNO, CNO, GRADE)

学生选课数据库



多粒度封锁模式

Microsoft SQL Server Management Studio (SQL Server 12.0.2000 - SA)

对象资源管理器

- AA\SQLEXPRESS (SQL Server 12.0.2000 - SA)
 - 数据库
 - 系统数据库
 - student_course
 - TMS
 - TRANSACTIONS
 - 数据库关系图
 - 表
 - 系统表
 - fileTables
 - dbo.Accounts
 - dbo.SC
 - 视图
 - 同义词
 - 可编程性
 - Service Broker
 - 存储
 - 安全性
 - 学生选课
 - 学生选课2
 - 学生选课3
 - 安全性
 - 服务器对象
 - 复制
 - 管理

事务 T1.sql - AA\SQLEXPRESS (SA (52))

```
BEGIN TRAN T1
SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';
```

100 %

	sno	grade
1	S01	79.0
2	S01	86.0

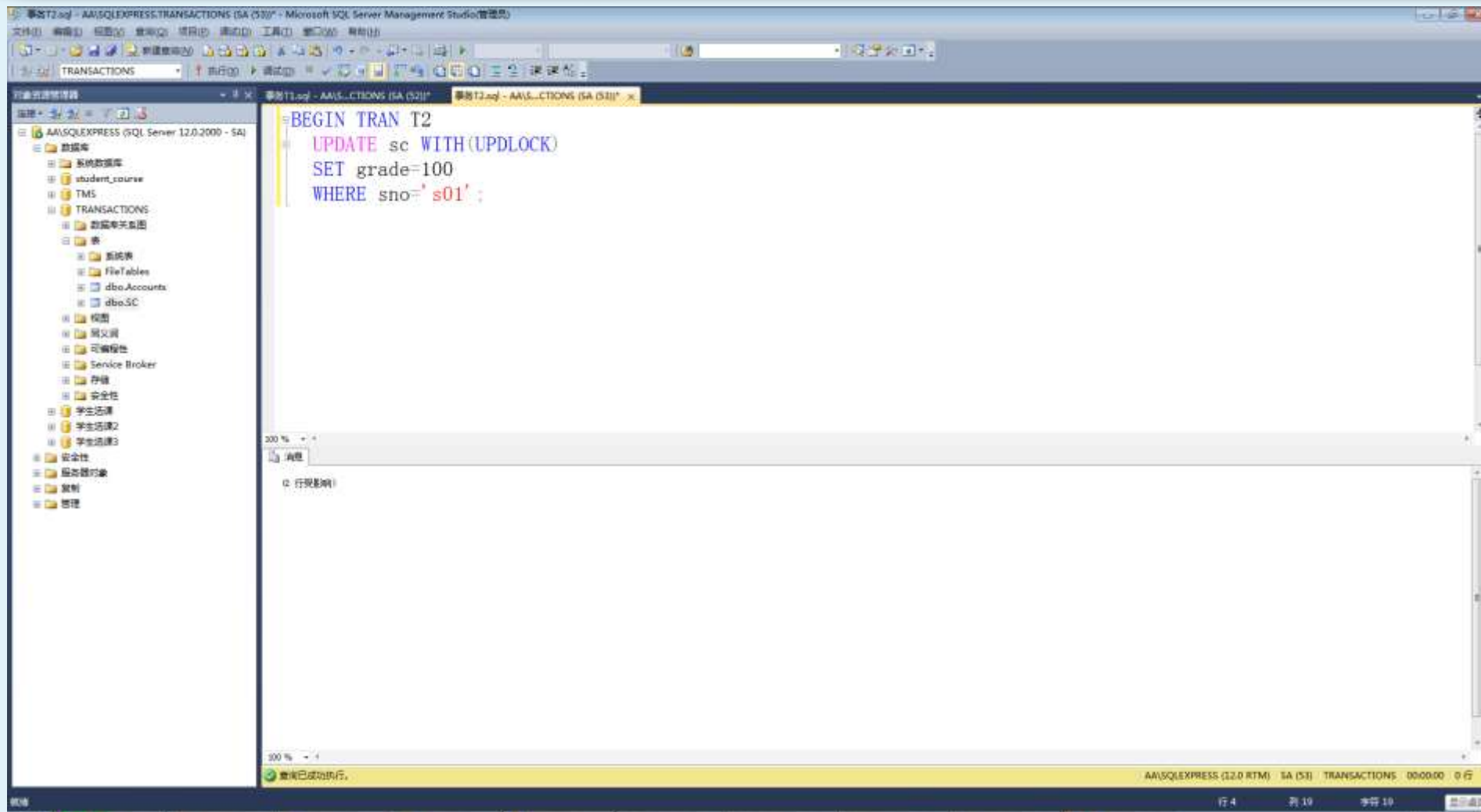
查询已成功执行。

AA\SQLEXPRESS (12.0 RTM) SA (52) TRANSACTIONS 00:00:00 2 行

行 4 列 19 字符 19



多粒度封锁模式





多粒度封锁模式

Microsoft SQL Server Management Studio (SSMS) interface showing a SQL query and its results.

Query:

```
BEGIN TRAN T1
SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (NOLOCK)
WHERE sno='s01';
```

Results:

sno	grade
1	100.0
2	100.0

查询已成功执行。



多粒度封锁模式

Microsoft SQL Server Management Studio (管理员)

TRANSACTIONS

对象资源管理器

AA\SQLEXPRESS (SQL Server 12.0.2000 - SA)

- 数据库
- 系统数据库
- student_course
- TMS
- TRANSACTIONS
 - 数据库关系型
 - 表
 - 系统表
 - FileTables
 - dbo.Accounts
 - dbo.SC
 - 视图
 - 同义词
 - 可编程性
 - Service Broker
 - 存储
 - 安全性
 - 学生选课
 - 学生选课2
 - 学生选课3
 - 安全性
 - 服务器对象
 - 复制
 - 管理

事务T1.sql - AA\S...CTIONS (SA (52))

```
BEGIN TRAN T1
SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (NOLOCK)
WHERE sno='s01';

UPDATE sc WITH (UPDLOCK)
SET grade=99
WHERE sno='s02';
```

100 %

消息

0 行受影响

100 %

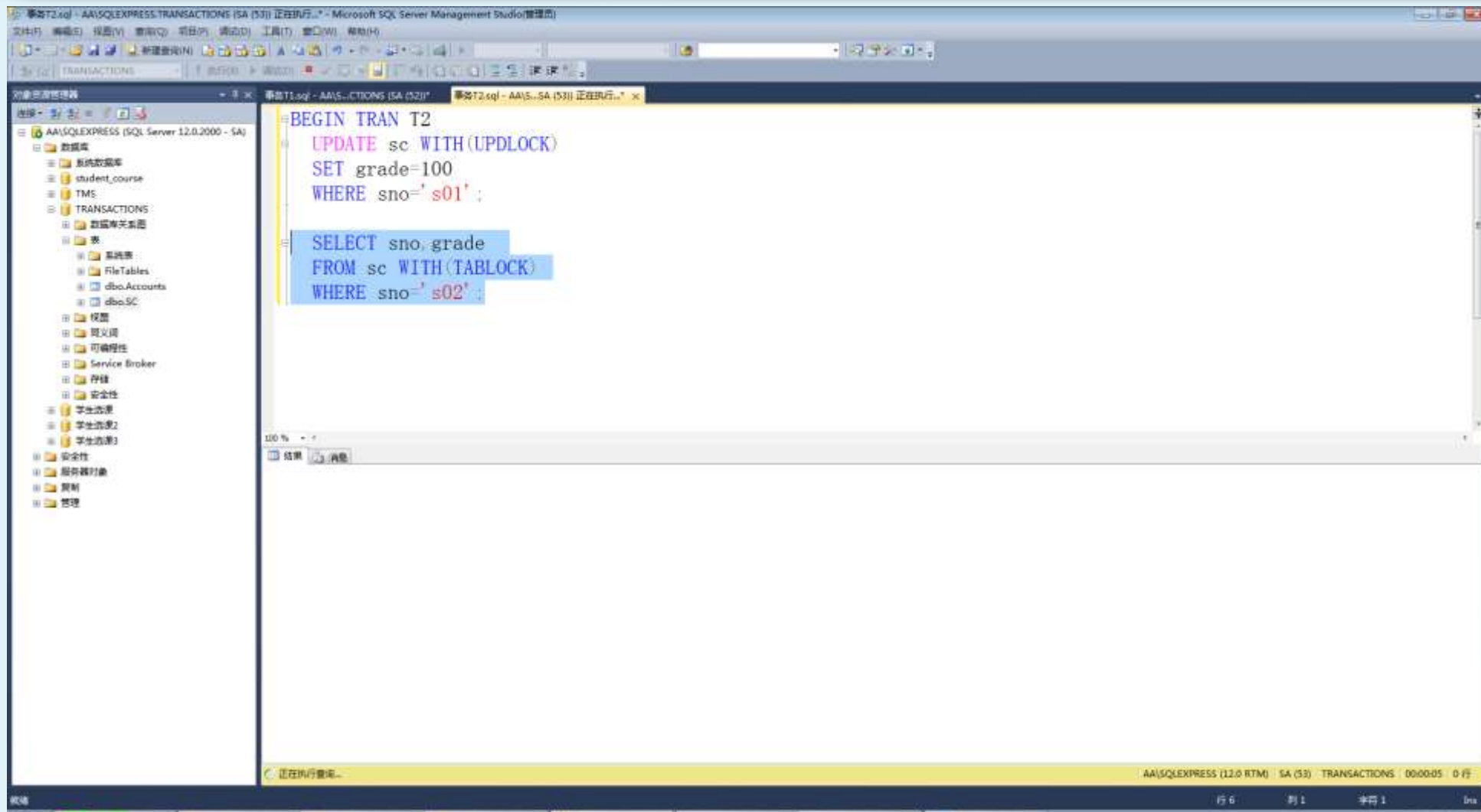
查询已成功执行。

AA\SQLEXPRESS (12.0 RTM) | SA (52) | TRANSACTIONS | 00:00:00 | 0 行

行 10 列 1 字符 1 Ins



多粒度封锁模式





多粒度封锁模式

Transaction T1 (Left Window):

```
SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (NOLOCK)
WHERE sno='s01';

UPDATE sc WITH (UPDLOCK)
SET grade=99
WHERE sno='s02';

COMMIT TRAN T1;
```

Transaction T2 (Right Window):

```
BEGIN TRAN T2

UPDATE sc WITH (UPDLOCK)
SET grade=100
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s02';
```

Status Bar (Bottom):

AA\SQLSERVER (12.0 RTM) - SA (DB) - TRANSACTIONS - 00:00:00 - 1 行



多粒度封锁模式

The screenshot displays the Microsoft SQL Server Management Studio interface. The left pane shows the database structure for 'AA\SQLEXPRESS (SQL Server 12.0.2000 - SA)'. The central pane shows a SQL script with the following queries:

```
UPDATE sc WITH (UPDLOCK)
SET grade=100
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s02';

ROLLBACK;

SELECT sno, grade
FROM sc
WHERE sno='s01';
```

The bottom pane shows the results of the queries, displaying a table with two rows:

sno	grade
1	S01 79.0
2	S01 86.0

The status bar at the bottom indicates the query was successful: '查询已成功执行。' (Query executed successfully.)



多粒度封锁模式

多粒度封锁模式实例

- NOLOCK: 读数据前不加锁（仅用于SELECT语句）。
- TABLOCK: 在表上加共享锁，在读完数据后立即释放锁。
- HOLDLOCK: 保持锁，用于TABLOCK后，将共享锁保留到事务完成。
- UPDLOCK: 在满足条件的元组上加更新锁。
- TABLOCKX: 在表上加排他锁。

会脏读

短期读锁
避免脏读

长期写锁
元组锁



多粒度封锁模式

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the server hierarchy for 'AA\SQLEXPRESS (SQL Server 12.0.2000 - SA)'. The right pane shows a SQL script in a query window:

```
BEGIN TRAN T1  
SELECT sno, grade  
FROM sc WITH (TABLOCK)  
WHERE sno='s01';
```

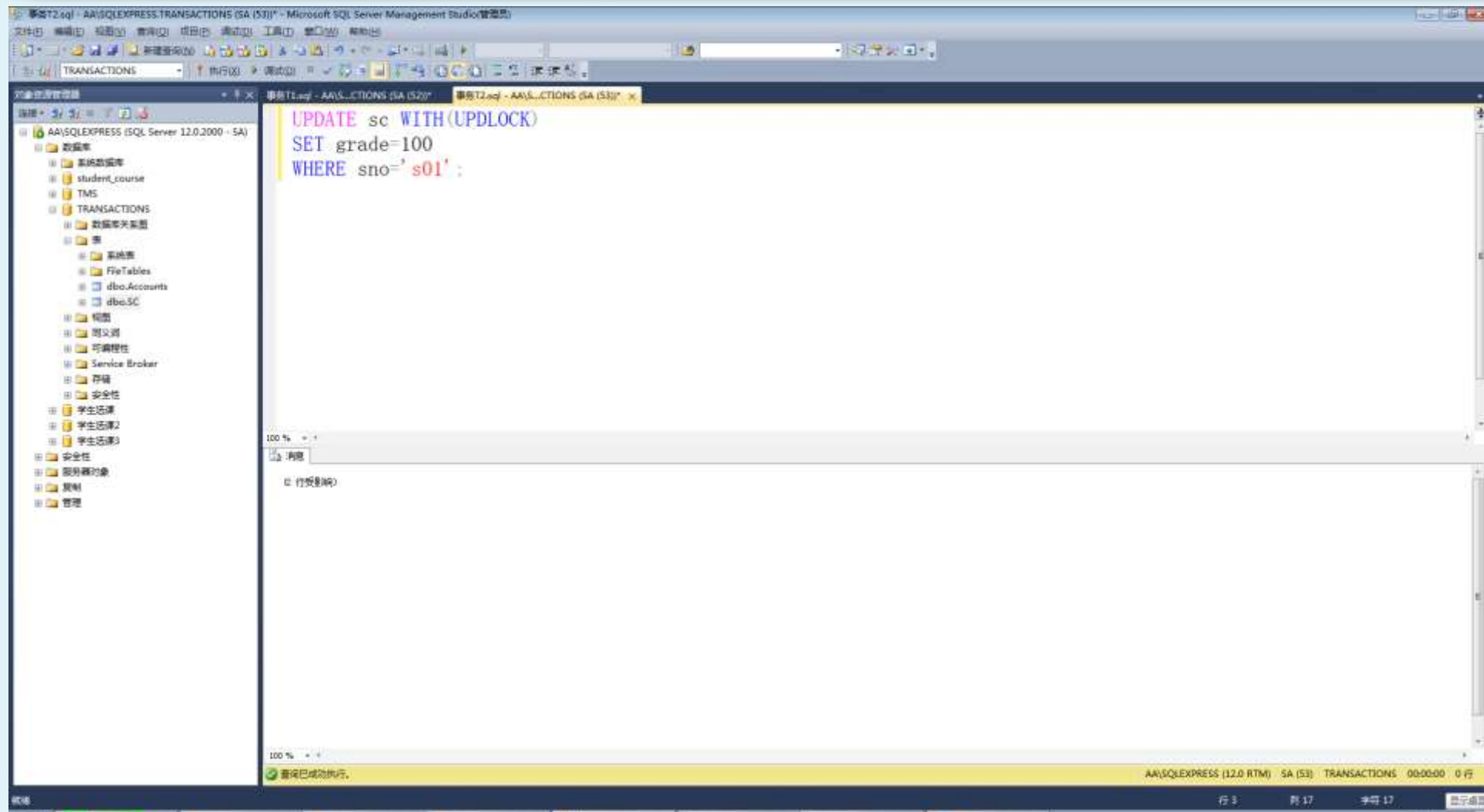
Below the script, the 'Results' pane shows the output of the query:

sno	grade
1	79.0
2	88.0

The status bar at the bottom indicates the query was executed successfully on the 'TRANSACTIONS' database, returning 2 rows in 0.00 seconds.

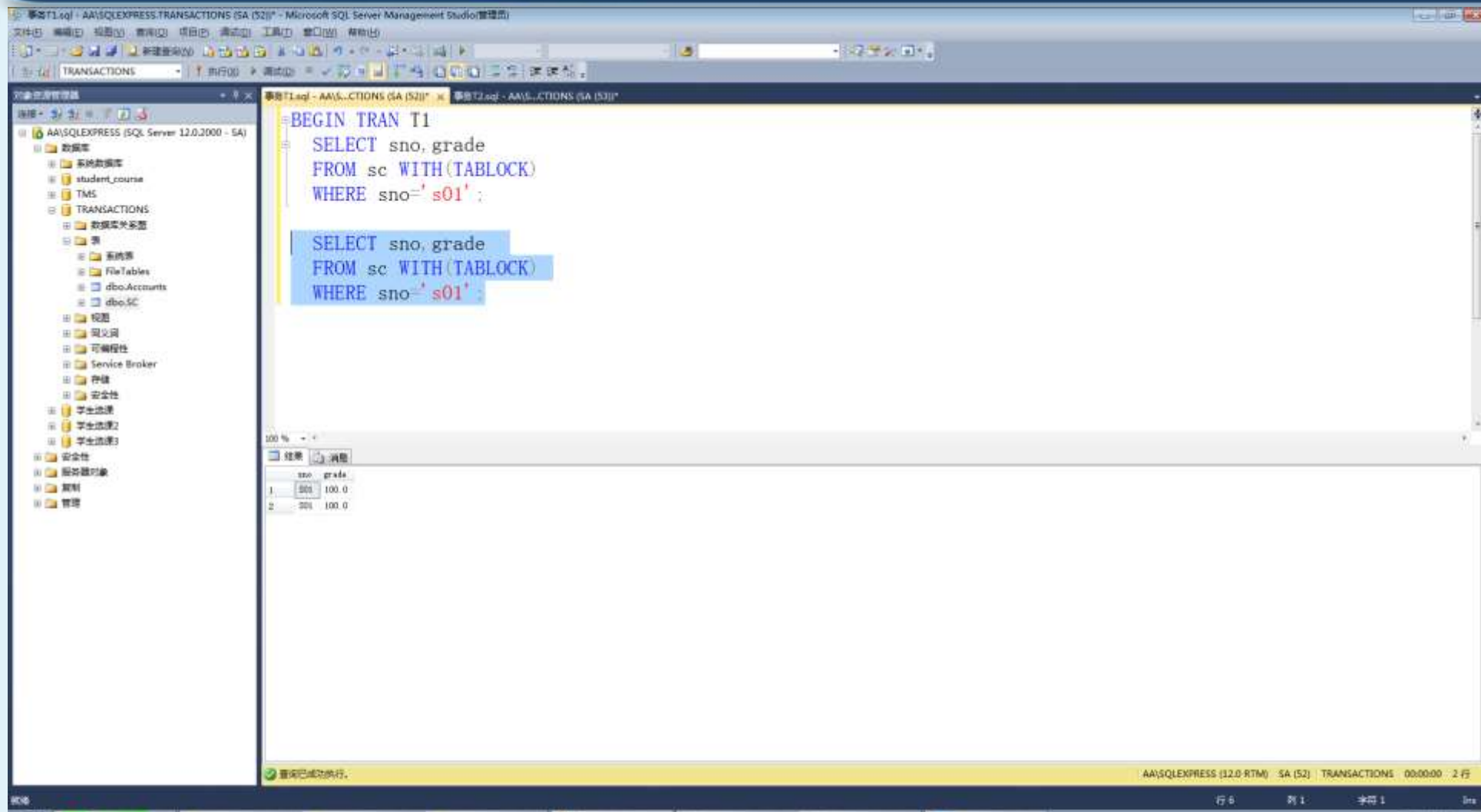


多粒度封锁模式





多粒度封锁模式





多粒度封锁模式

Microsoft SQL Server Management Studio (SQL Server 12.0.2000 - SA)

事务 T1.sql - AA\SQLEXPRESS.TRANSACTIONS (SA (52))

```
BEGIN TRAN T1
SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (TABLOCK HOLDLOCK)
WHERE sno='s01';
```

结果

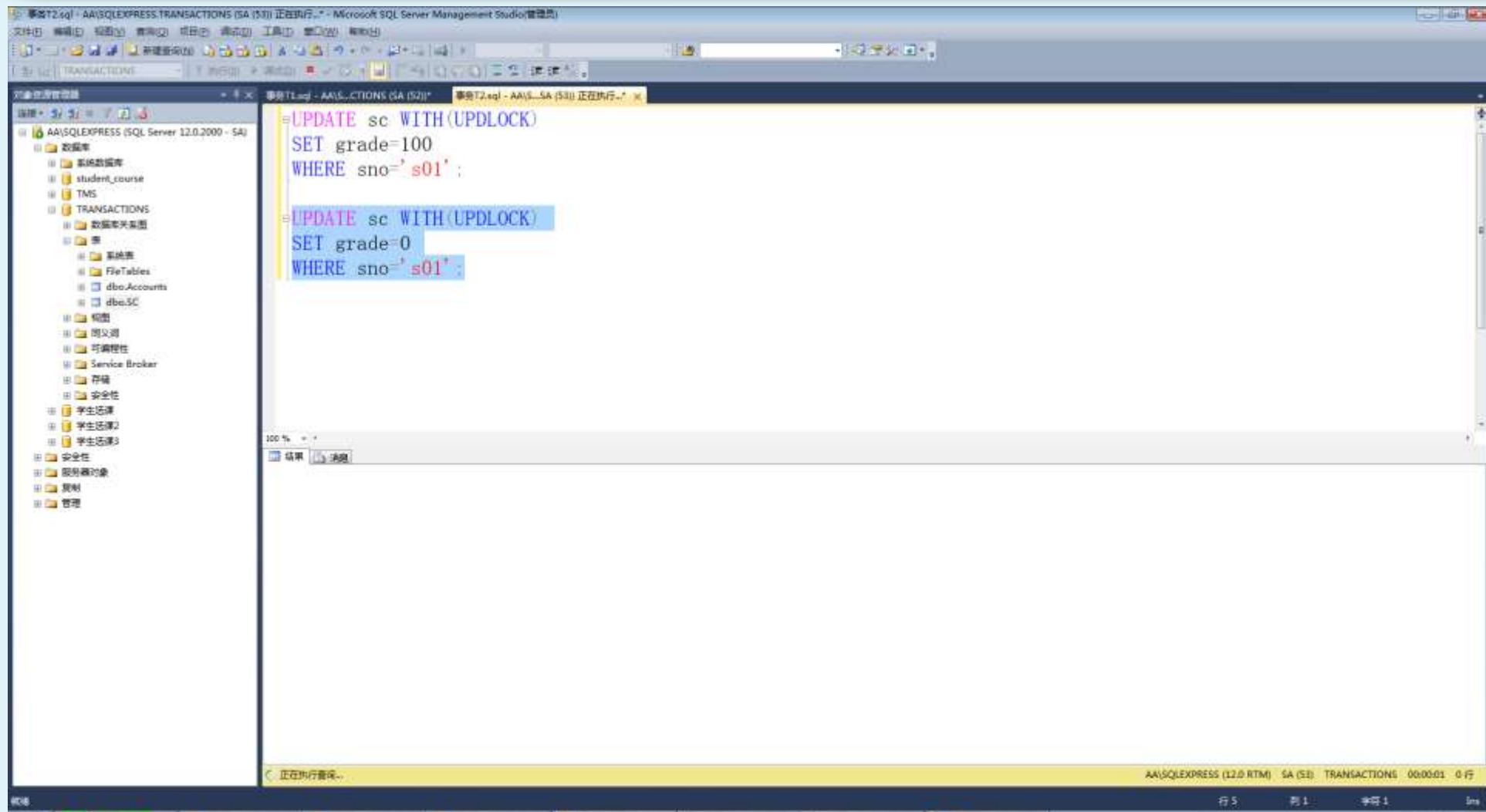
sno	grade
s01	100.0
s01	100.0

查询已成功执行。

AA\SQLEXPRESS (12.0 RTM) : SA (52) : TRANSACTIONS : 00:00:00 : 2 行



多粒度封锁模式





多粒度封锁模式

WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (TABLOCK HOLDLOCK)
WHERE sno='s01';

SELECT sno, grade
FROM sc WITH (TABLOCK HOLDLOCK)
WHERE sno='s01';

sno	grade
s01	100.0

查询已成功执行。

AAVSQLEXPRESS (12.0 RTM) - SA (52) - TRANSACTIONS : 00:00:00 2 行



多粒度封锁模式

Transaction T1 (Transaction 1):

```
SELECT sno, grade
FROM sc WITH (TABLOCK)
WHERE sno='s01';

COMMIT TRAN T1
```

Transaction T2 (Transaction 2):

```
UPDATE sc WITH (UPDLOCK)
SET grade=100
WHERE sno='s01';

UPDATE sc WITH (UPDLOCK)
SET grade=0
WHERE sno='s01';
```

The screenshot shows the SQL Server Management Studio interface with the 'TRANSACTIONS' database selected. The left pane shows the database structure, including the 'sc' table. The right pane shows the execution of the transactions. The status bar at the bottom indicates that both transactions completed successfully.



多粒度封锁模式

多粒度封锁模式实例

- NOLOCK: 读数据前不加锁（仅用于SELECT语句）。
- TABLOCK: 在表上加共享锁，在读完数据后**立即释放**锁。
- HOLDLOCK: 保持锁，用于TABLOCK后，将共享锁保留到**事务完成**。
- UPDLOCK: 在满足条件的元组上加更新锁。
- TABLOCKX: 在表上加排他锁。

避免脏读
不可重复读

长期读锁
可重复读



多粒度封锁模式

多粒度封锁的实现

- DBMS为解决多粒度封锁技术中细粒度锁和粗粒度锁之间的冲突，会在细粒度对象加锁前在粗粒度对象上加**意向锁**。
- 意向锁的作用就是标识关系中的某些元组正在被锁定或其他用户将要锁定关系中的某些元组。
- 意向锁是由系统隐式进行添加的，不能人为干预。
- 意向锁有**意向共享锁 (IS)**、**意向排他锁 (IX)** 之分。






多粒度封锁模式

多粒度封锁的实现

		事务T2申请的锁					
		T2	S	X	IS	IX	—
事务T1获得的锁	T1						
	S	Y	N	Y	N	Y	
	X	N	N	N	N	Y	
	IS	Y	N	Y	Y	Y	
	IX	N	N	Y	Y	Y	
	—	Y	Y	Y	Y	Y	



小结

-  多粒度封锁技术可提高并发事务的并发程度。
-  用户可根据应用需求，选择封锁粒度，显式地为事务中的操作加锁，控制事务的并发执行。
-  多粒度封锁的灵活应用会带来数据的不一致问题。