

试验十六 事务

目的:1 掌握并理解事务

一 理解 rollback

1 在查询分析器输入下列语句并执行，记录该学生的年龄。

```
Select * from student where sno=' 0001'
```

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	20	JSJ

2 执行下列 语句序列 A:

```
BEGIN TRANsaction
```

```
Update student set sage=sage+1 where sno=' 0001'
```

```
Select * from student where sno=' 0002'
```

	Sno	Sname	Ssex	Sage	Sdept
1	0002	李四	女	25	SX

此事务结束了吗？

未结束, 没有 end

3 执行:

```
Select * from student where sno=' 0001'
```

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	21	JSJ

记录该学生的年龄。

思考: student 中的 0001 的年龄确实被更改了吗? 为什么?

被更改了, 因为在事务中 0001 的年龄以及更改了

4 执行下列语句。

```
ROLLBACK TRANsaction
```

然后再执行:

```
Select * from student where sno=' 0001'
```

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	20	JSJ

， 观察 0001 的年龄，解释发生这种现象的原因。

事务回滚, 因此 0001 的年龄回到了修改之前

二 理解 commit

1 在查询分析器输入下列语句并执行，记录该学生的年龄。

```
Select * from student where sno=' 0001'
```

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	20	JSJ

2 执行下列 语句序列 A:

```
BEGIN TRANsaction
```

```
Update student set sage=sage+1 where sno=' 0001'
```

```
Select * from student where sno=' 0002'
```

	Sno	Sname	Ssex	Sage	Sdept
1	0002	李四	女	25	SX

3 执行:

```
commit transaction
```

```
Select * from student where sno=' 0001'
```

记录结果, 此时更改后的数据被永久保存了吗?

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	21	JSJ

被永久保存了

三 执行下列 语句序列

```
BEGIN TRANsaction
```

```
Update student set sage=sage+1 where sno=' 0001'
```

```
Update sc set grade=grade + 1 where sno=' 0002' and cno=' 1001'
```

```
Rollback
```

上述指令执行后, 数据库发生了什么变化?

无变化,因为变化被回滚了

试验十七 锁

目的:1 理解锁的概念及锁的作用

一 利用帮助系统了解 Sql-server 的下列语句的含义

1 锁的隔离级别

```
SET TRANSACTION ISOLATION LEVEL Serializable
```

这条语句用于设置数据库事务的隔离级别为“Serializable”(可串行化)。隔离级别是用来控制并发访问数据库时事务之间的隔离程度。“Serializable”是最高的隔离级别,它确保事务完全串行化执行,避免了脏读、不可重复读和幻读的问题。在该隔离级别下,事务会对读取的数据进行锁定,直到事务结束。

2 设置锁定超时时间

```
SET LOCK_TIMEOUT 5000
```

这条语句用于设置锁定超时时间为 5000 毫秒(5 秒)。当一个事务请求获取锁定资源时,如果在指定的超时时间内无法获取到锁定,系统会将该请求取消,并返回一个错误。这可以用来控制事务在等待锁定资源时的等待时间。

3 SP_LOCK

这是一个系统存储过程，用于获取当前数据库中的锁定信息。它返回一个结果集，包含了当前活动的锁定信息，如锁定的资源、锁定的类型、持有该锁定的事务等。通过执行该存储过程，可以查看数据库中的锁定情况，帮助进行故障排查或性能优化。

二 观察封锁

1 执行语句序列 A

```
BEGIN TRANsaction
```

```
Update student set sage=sage+1 where sno=' 0001'
```

```
Select * from student where sno=' 0002'
```

2 在查询分析器中打开第二个连接(连接 school)[文件-连接]，输入下列语句：

1) select * from student where sno=' 0002'

结果		消息			
	Sno	Sname	Ssex	Sage	Sdept
1	0002	李四	女	25	SX

记录执行结果,说明原因。

2) select * from student where sno=' 0001'

记录执行结果,说明原因。(如上一步没有停止，则强行终止)

如果在步骤一中的第二个连接还在运行，将无法获得学生 '0001' 的最新数据，因为第一个连接的事务还未提交，对学生 '0001' 的更新操作尚未生效。

如果强行终止步骤一中的第二个连接，然后在步骤二中执行查询语句，将获得学生 '0001' 的最新数据，包括之前更新的年龄加 1 的结果。

3) update student set sname=' aaa' where sno=' 0002'

记录执行结果,说明原因。(如上一步没有停止，则强行终止)

如果在之前的步骤中已经执行了该事务且事务未提交，则更新语句将会阻塞，等待事务完成或者超时。

如果之前的事务已提交或者没有进行任何事务操作，那么更新语句将立即执行。

如果更新语句成功执行，将会将学生表中 "sno" 为 '0002' 的学生的 "sname" (姓名) 字段更新为 'aaa'。

如果更新语句受阻塞，等待事务完成后才能执行，或者超过了锁定超时时间（如果已设置锁定超时时间），那么更新语句将失败，不会进行任何更改。

4) 强行终止上一步的命令，然后执行语句：

```
DBCC opentran
```

(1 行受影响)

数据库 'school' 的事务信息。

最早的活动事务：

SPID (服务器进程 ID): 51

UID (用户 ID): -1

名称 : user_transaction

LSN : (41:1064:6)

开始时间 : 06 4 2023 7:04:41:380PM

SID : 0x0105000000000000515000000427fd56c52488121cb956758e9030000

DBCC 执行完毕。如果 DBCC 输出了错误信息，请与系统管理员联系。

完成时间: 2023-06-04T19:04:50.6998834+08:00

记录结果，思考：如何知道此事务是那一台计算机发出的？

5) 执行:

```
select * from student where sno=' 0001'
```

记录执行结果, 说明原因

语句堵塞, 无法执行

然后回到**第一个连接**中, 执行语句:

```
commit Tran
```

观察并记录第二个连接窗口中的现象, 说明原因

消息					
	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	22	JSJ

三 了解锁的类型

1 执行下列语句

```
BEGIN TRAN
```

```
Select * from student where sno=' 0001'
```

```
Print 'server process ID (spid) : '
```

```
Print @@spid
```

1) 然后执行下列语句

```
exec sp_lock
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	51	9	0	0	DB		S	GRANT
2	51	1	1787153412	0	TAB		IS	GRANT
3	51	32767	-571204656	0	TAB		Sch-S	GRANT
4	66	9	0	0	DB		S	GRANT

注意根据事务中输出的 spid , 观察结果中相应 spid 的记录, 观察加锁。

2) 然后执行下列语句

```
commit tran
```

```
exec sp_lock
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	51	9	0	0	DB		S	GRANT
2	51	1	1787153412	0	TAB		IS	GRANT
3	51	32767	-571204656	0	TAB		Sch-S	GRANT
4	66	9	0	0	DB		S	GRANT

注意根据事务中输出的 spid , 观察结果中相应 spid 的记录, 观察加锁。

2 执行下列语句

```
BEGIN TRAN
```

```
Update student set sage=sage + 1 where sno=' 1001'
```

```
Print 'server process ID (spid) : '
```

```
Print @@spid
```

(0 行受影响)

```
server process ID (spid) :  
51
```

完成时间: 2023-06-04T19:23:52.4997677+08:00

1) 然后执行下列语句

```
exec sp_lock
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	51	9	1205579333	1	PAG	1:312	IX	GRANT
2	51	9	1205579333	0	TAB		IX	GRANT
3	51	1	1787153412	0	TAB		IS	GRANT
4	51	32767	-571204656	0	TAB		Sch-S	GRANT
5	51	9	0	0	DB		S	GRANT
6	66	9	0	0	DB		S	GRANT

注意根据事务中输出的 spid , 观察结果中相应 spid 的记录, 观察加锁。

2) 然后执行下列语句

```
commit tran  
exec sp_lock
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	51	9	0	0	DB		S	GRANT
2	51	1	1787153412	0	TAB		IS	GRANT
3	51	32767	-571204656	0	TAB		Sch-S	GRANT
4	66	9	0	0	DB		S	GRANT

注意根据事务中输出的 spid , 观察结果中相应 spid 的记录, 观察加锁。

3 使用 SET TRANSACTION ISOLATION LEVEL Serializable

然后重新执行 三和四步, 观察与原来有何不同。

4 了解表级锁 (查看帮助文件)

```
BEGIN TRAN  
Select * from student (TABLOCKX) where sno=' 1002'  
Print 'Server Process ID (spid): '  
Print @@spid
```

然后执行:

```
exec sp_lock
```

注意根据事务中输出的 spid , 观察结果中相应 spid 的记录, 观察加锁类型。

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	51	9	1205579333	0	TAB		X	GRANT
2	51	9	1205579333	0	TAB		X	GRANT
3	51	9	1205579333	0	TAB		X	GRANT
4	51	9	1205579333	0	TAB		X	GRANT
5	51	9	1205579333	0	TAB		X	GRANT
6	51	9	1205579333	0	TAB		X	GRANT
7	51	9	1205579333	0	TAB		X	GRANT
8	51	9	1205579333	0	TAB		X	GRANT
9	51	9	1205579333	0	TAB		X	GRANT
10	51	9	1205579333	0	TAB		X	GRANT
11	51	9	1205579333	0	TAB		X	GRANT
12	51	9	1205579333	0	TAB		X	GRANT
13	51	9	1205579333	0	TAB		X	GRANT
14	51	9	1205579333	0	TAB		X	GRANT
15	51	9	1205579333	0	TAB		X	GRANT
16	51	9	1205579333	0	TAB		X	GRANT
17	51	1	1787153412	0	TAB		IS	GRANT
18	51	32767	-571204656	0	TAB		Sch-S	GRANT
19	51	9	0	0	DB		S	GRANT
20	66	9	0	0	DB		S	GRANT

5 了解锁定超时

a) 执行下列语句，设置锁定超时为 1000 ms

```
set lock_timeout 1000
go
BEGIN TRAN
    Select * from student (TABLOCKX) where sno=' 1002'
```

Sno	Sname	Ssex	Sage	Sdept
-----	-------	------	------	-------

2) 打开第二个连接

执行：

```
select * from student
记录观察到的现象。
```

语句堵塞,无法执行

3) 在打开的第二个连接中

```
set lock_timeout 10000
go
select * from student
```

记录观察到的现象。

语句堵塞,无法执行

试验十八 恢复技术

- 目的：1 掌握数据库的备份及恢复的方法。
2 了解备份方案的设定

一 完全备份的建立与恢复

1 建立完全备份

```
USE school
GO
BACKUP DATABASE school TO DISK=' C:\schooldata.bak'
```

已为数据库 'school', 文件 'school' (位于文件 1 上)处理了 640 页。
已为数据库 'school', 文件 'school_log' (位于文件 1 上)处理了 2 页。
BACKUP DATABASE 成功处理了 642 页, 花费 0.018 秒(278.428 MB/秒)。

2 查看备份文件中的信息

```
RESTORE FILELISTONLY FROM DISK=' c:\schooldata.bak'
RESTORE HEADERONLY FROM DISK=' c:\schooldata.bak'
```

消息														
	LogicalName	PhysicalName	Type	FileGroupName	Size	MaxSize	FileId	CreatedLSN	DroppedLSN	Uniquelfid	ReadonlyLSN	ReadFritelLSN	BackupSizeInBytes	SourceFile
1	school	D:\SQL\SQL_server\MSSQL16_SQLEXPRESS\MSSQL\DATA...	D	PRIMARY	8388608	35184372080640	1	0	0	1B093FC2-17EA-45ED-988D-BB198ACD4477	0	0	5111808	4096
2	school_log	D:\SQL\SQL_server\MSSQL16_SQLEXPRESS\MSSQL\DATA...	L	NULL	8388608	2199023255552	2	0	0	CE11F5D0-4501-41B2-9F1E-A535F9AA1CCEA	0	0	0	4096

3 恢复完全备份

1) 先删除数据库 School

```
USE Master
GO
DROP DATABASE school
```

2) 然后恢复.

```
RESTORE DATABASE school from DISK=' c:\schooldata.bak'
```

已为数据库 'school', 文件 'school' (位于文件 1 上)处理了 640 页。
已为数据库 'school', 文件 'school_log' (位于文件 1 上)处理了 2 页。
RESTORE DATABASE 成功处理了 642 页, 花费 0.009 秒(556.857 MB/秒)。

完成时间：2023-06-04T19:33:03.6459702+08:00

3): 查看 school 的 student 中的数据

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	22	JSJ
2	0002	aaa	女	25	SX
3	0701	刘欢	男	26	SX
4	0702	赵欢	男	31	SX
5	4001	赵茵	男	20	SX
6	4002	杨华	女	21	NULL
7	5001	赵强	男	22	SX
8	5002	杨丽华	女	21	JSJ
9	5003	李静	女	22	SX

二 建立差异备份

1 建立备份

- 1) 制作数据文件备份 schoolDiff.bak
- 2) 把学号 7001, 姓名: 王海, 性别: 男, 年龄为 23 的学生加入 student
- 3) 制作 school 的差异备份, 存入 schoolDiff.bak

```
BACKUP DATABASE school TO DISK='schoolDiff.bak' WITH DIFFERENTIAL
```

- 4) 把学号 7002, 姓名: 赵燕, 性别: 女, 年龄为 22 的学生加入 student

- 5) 制作 school 的差异备份, 存入 schoolDiff.bak

```
BACKUP DATABASE school TO DISK='schoolDiff.bak' WITH DIFFERENTIAL
```

2 查看备份文件 schoolDiff.bak 中的信息

3 删除 school 数据库

4 恢复数据库 school 到第 2 步状态

```
RESTORE DATABASE school from DISK=' c:\schoolDiff.bak' WITH file=1 NORECOVERY
```

```
RESTORE DATABASE school from DISK=' c:\schoolDiff.bak' WITH file=2
```

```
Select * from student
```

观察 student 数据

5 恢复数据库 school 到最新状态

```
RESTORE DATABASE school from DISK=' c:\schoolDiff.bak' WITH file=1 NORECOVERY
```

```
RESTORE DATABASE school from DISK=' c:\schoolDiff.bak' WITH file=3
```

```
Select * from student
```

观察 student 数据

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	22	JSJ
2	0002	aaa	女	25	SX
3	0701	刘欢	男	26	SX
4	0702	赵欢	男	31	SX
5	4001	赵茜	男	20	SX
6	4002	杨华	女	21	NULL
7	5001	赵强	男	22	SX
8	5002	杨丽华	女	21	JSJ
9	5003	李静	女	22	SX

思考: 如果仅执行下述恢复语句,能查看 student 的数据吗?

```
RESTORE DATABASE school from DISK=' c:\schoolDiff.bak' WITH file=1 NORECOVERY
```

Select * from student

三 利用日志备份

1 设置故障还原模型为: 完全



2 建立备份

1) 制作数据文件备份 schooldata1.bak

```
use school
go
backup database school to disk ='E:\schooldata1.bak'
```

00 % ▾

消息

已为数据库 'school', 文件 'school1' (位于文件 1 上)处理了 1000 页。
已为数据库 'school', 文件 'school1_log' (位于文件 1 上)处理了 1 页。
BACKUP DATABASE 成功处理了 1001 页, 花费 0.029 秒(269.447 MB/秒)。

2) 把学号 7003, 姓名: 王江, 性别: 男, 年龄为 23 的学生加入 student

```
insert into s(sno, sname, ssex, sage) values('7003', '王江', '男', 23)
```

0 % ▾

消息

(1 行受影响)

2) 制作日志备份存入 schoollog.bak

```
use school
go
backup database school to disk = 'E:\schoollog.bak'
```

100 %

消息

已为数据库 'school', 文件 'school1' (位于文件 1 上)处理了 1000 页。
已为数据库 'school', 文件 'school1_log' (位于文件 1 上)处理了 1 页。
BACKUP DATABASE 成功处理了 1001 页, 花费 0.039 秒(200.358 MB/秒)。

3) 把学号 7004, 姓名: 赵兰, 性别: 女, 年龄为 22 的学生加入 student

```
insert into s(sno, sname, ssex, sage) values('7004', '赵兰', '女', 22)
```

100 %

消息

(1 行受影响)

4) 制作日志备份存入 schoollog.bak

```
use school
go
backup database school to disk ='E:\schoollog.bak'
```

100 %

消息

已为数据库 'school', 文件 'school1' (位于文件 2 上)处理了 1000 页。
已为数据库 'school', 文件 'school1_log' (位于文件 2 上)处理了 1 页。
BACKUP DATABASE 成功处理了 1001 页, 花费 0.031 秒(252.063 MB/秒)。

3 观察 schoollog 中的信息

```
RESTORE FILELISTONLY FROM DISK='E:\schoollog.bak'
RESTORE HEADERONLY FROM DISK='E:\schoollog.bak'
```

00 %

结果 消息

	LogicalName	PhysicalName	Type	FileGroupName	Size	MaxSize	FileId	CreateLSN	DropLSN	UniqueId	ReadOnl
1	school1	C:\Program Files\Microsoft SQL Server\MSSQL16.M...	D	PRIMARY	142606336	35184372080640	1	0	0	10AD0CF3-8825-4E28-8C59-2677E346CD6E	0
2	school1_log	C:\Program Files\Microsoft SQL Server\MSSQL16.M...	L	NULL	813694976	2199023255552	2	0	0	85B4F90C-472B-469B-8E05-056A0F5E884B	0

	BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate
1	NULL	NULL	1	NULL	0	1	2	DESKTOP-H43207N\DELL	DESKTOP-H43207N	school	957	2023-06-04 15:24:55.000
2	NULL	NULL	1	NULL	0	2	2	DESKTOP-H43207N\DELL	DESKTOP-H43207N	school	957	2023-06-04 15:24:55.000

4 删除 school 数据库

```
use master
go
drop database school
```

0 %

消息

命令已成功完成。

5 利用 schooldata1.bak 及 schoollog.bak 恢复数据库 school 到最新状态

```
RESTORE DATABASE school from DISK='E:\schoollog.bak' with file=1,norecovery ,replace
RESTORE DATABASE school from DISK='E:\schoollog.bak' with file=2
```

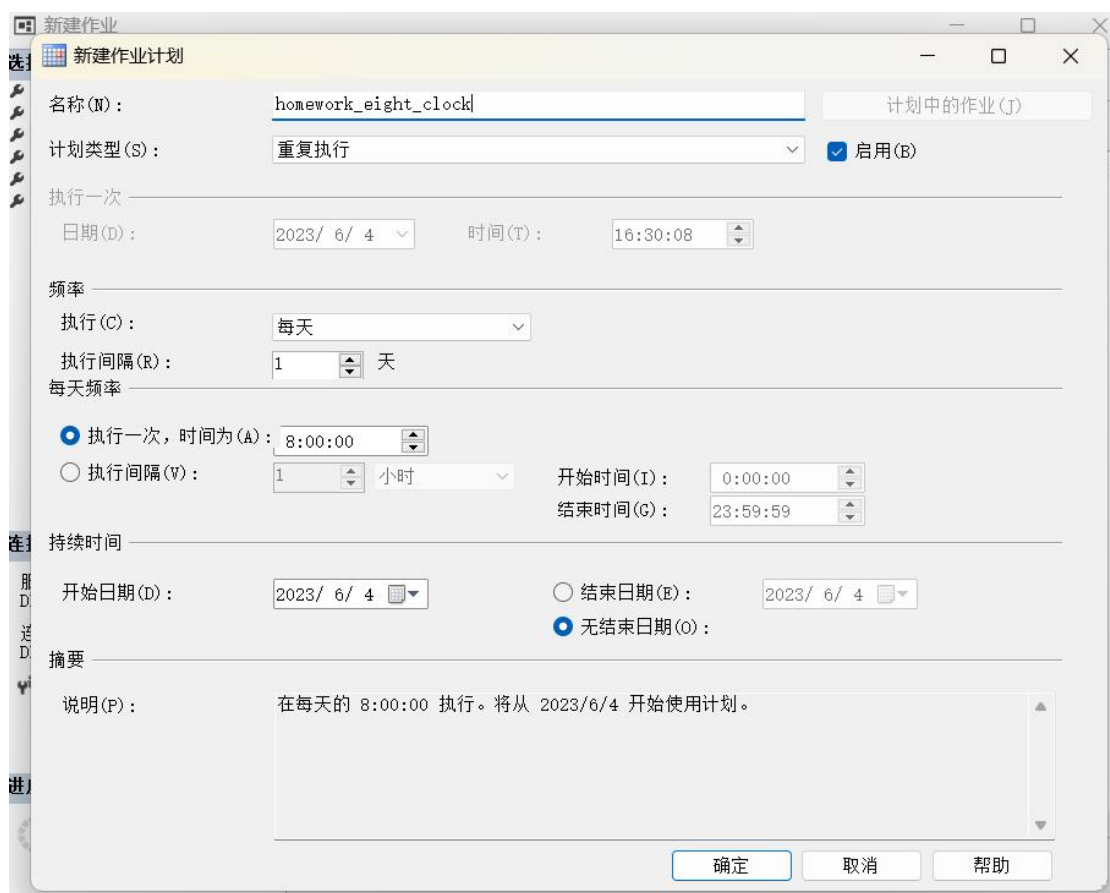
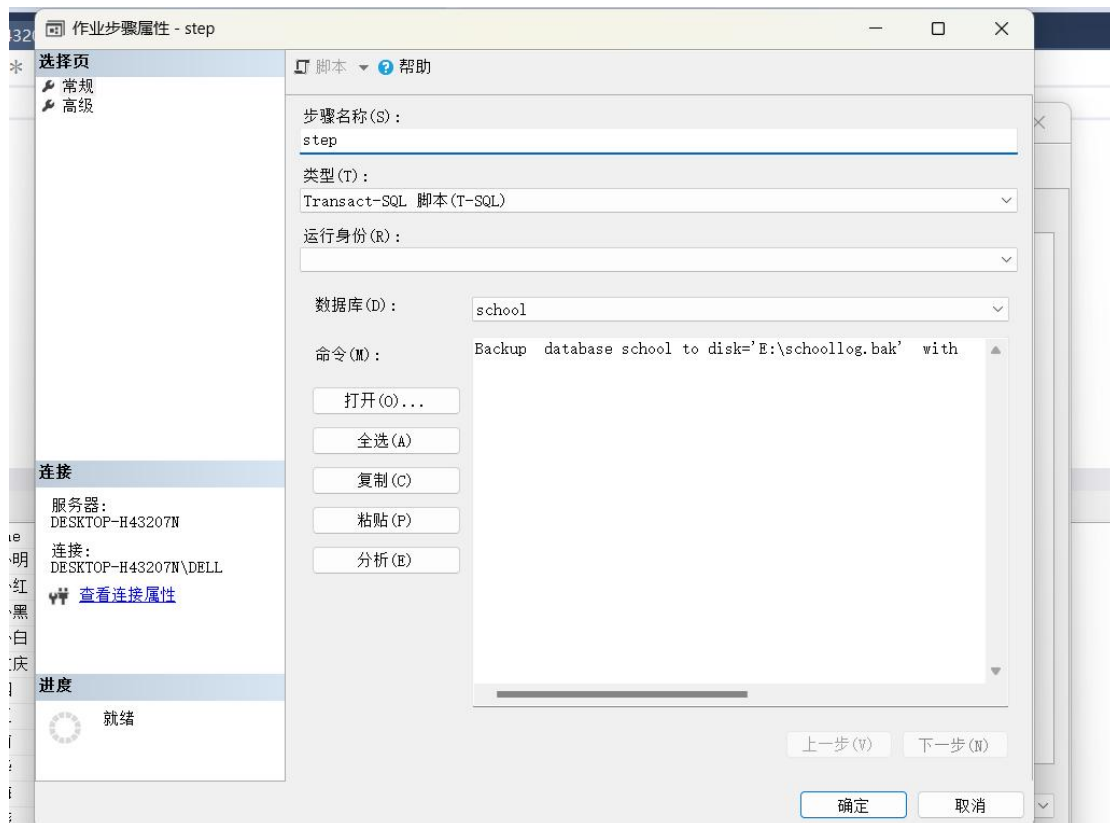
00 %

消息

已为数据库 'school', 文件 'school1' (位于文件 1 上)处理了 1000 页。
已为数据库 'school', 文件 'school1_log' (位于文件 1 上)处理了 1 页。
RESTORE DATABASE 成功处理了 1001 页, 花费 0.588 秒(13.289 MB/秒)。
已为数据库 'school', 文件 'school1' (位于文件 2 上)处理了 1000 页。
已为数据库 'school', 文件 'school1_log' (位于文件 2 上)处理了 1 页。
RESTORE DATABASE 成功处理了 1001 页, 花费 0.577 秒(13.542 MB/秒)。

四 使用企业管理器练习备份调度策略

- 1 对数据库 school 每天上午 8 时进行一次数据库完全备份



2 对数据库 school 的每隔 1 分钟备份进行一次差异备份。

作业步骤属性 - step1

选择项

常规

高级

连接

服务器:
DESKTOP-H43207N
连接:
DESKTOP-H43207N\DELL
[查看连接属性](#)

进度

就绪

脚本 帮助

步骤名称(S):
step1

类型(T):
Transact-SQL 脚本(T-SQL)

运行身份(R):

数据库(D):
school

命令(M):
backup database school to disk=N'E:\datadiff.bak' with diff

打开(O)...

全选(A)

复制(C)

粘贴(P)

分析(E)

上一步(V)

下一步(N)

确定

取消

作业计划属性 - diff_plan

名称(N):
diff_plan

计划中的作业(J)

计划类型(S):
重复执行

☒ 启用(E)

执行一次

日期(D):
2023/ 6/ 4

时间(T):
16:39:29

频率

执行(C):
每天

执行间隔(R):
1 天

每天频率

☐ 执行一次, 时间为(A):
0:00:00

☒ 执行间隔(V):
1 分钟

开始时间(I):
0:00:00

结束时间(G):
23:59:59

持续时间

开始日期(D):
2023/ 6/ 4

☐ 结束日期(E):
2023/ 6/ 4

☒ 无结束日期(O):

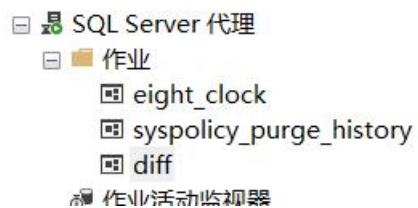
摘要

说明(P):
每天在 0:00:00 和 23:59:59 之间、每 1 分钟执行一次。将从 2023/6/4 开始使用计划。

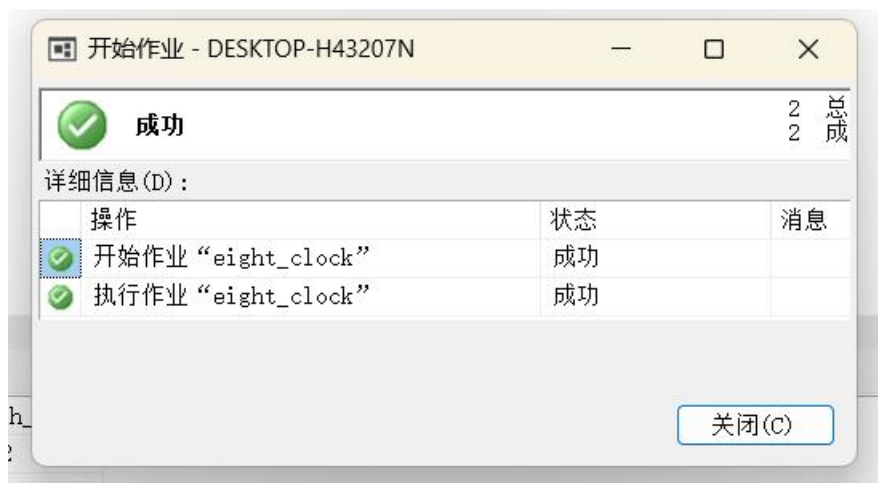
确定

取消

帮助



3 手工启动两个备份作业



5 删除 school 数据库

```
use master
go
drop database school
```

100 %

消息

命令已成功完成。

完成时间: 2023-06-04T16:42:59.9570310+08:00

```
select *
from s
```

100 %

结果 消息

	sno	sname	ssex	sage	sdept	birth_year
2	0002	张春华	女	21.0	SX	2001
3	001	陈小明	女	21.0	SX	2002
4	003	张小红	女	18.0	TST	1999

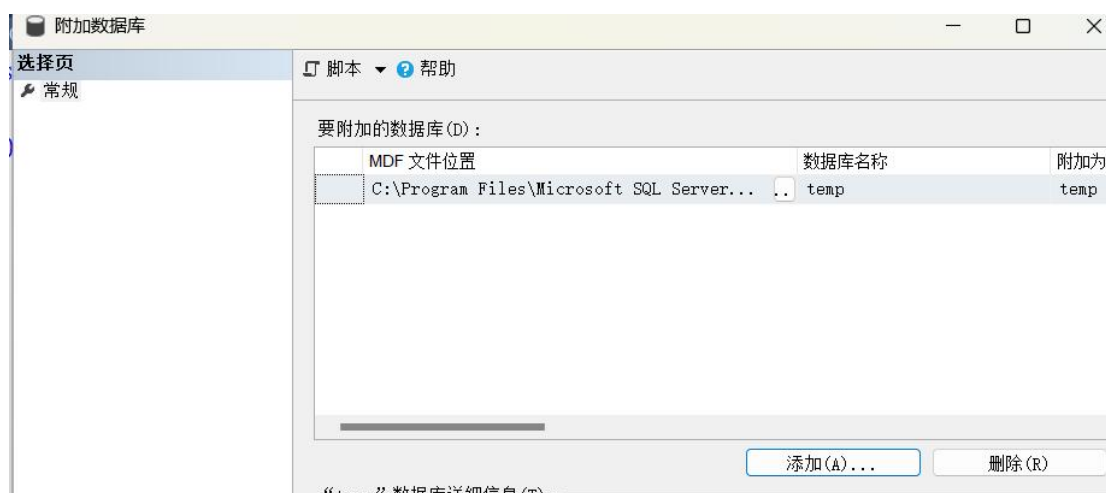
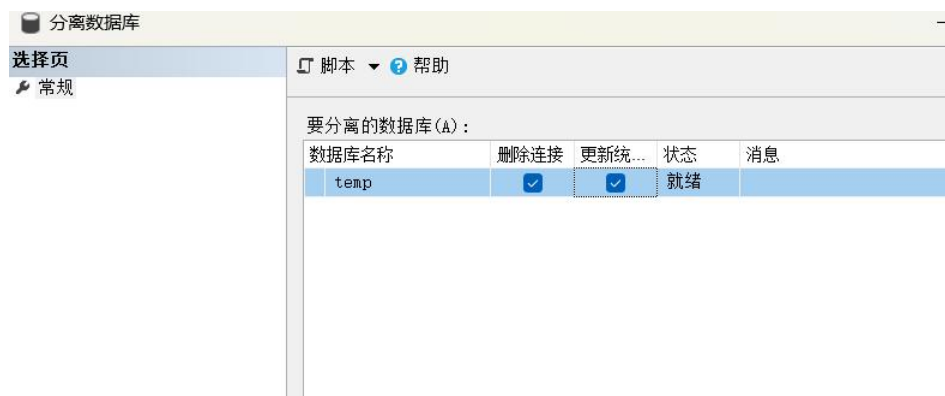
6 利用 1, 2 步的备份进行 school 的恢复。

思考：如何把备份文件备份到另外一台计算机上。

可以使用网络共享或 FTP 等文件传输协议，将备份文件从源计算机复制到目标计算机。

可以配置数据库服务器之间的数据库备份和复制策略，将备份文件自动复制到目标计算机上的指定位置。

五 使用企业管理器练习数据库的分离及附加

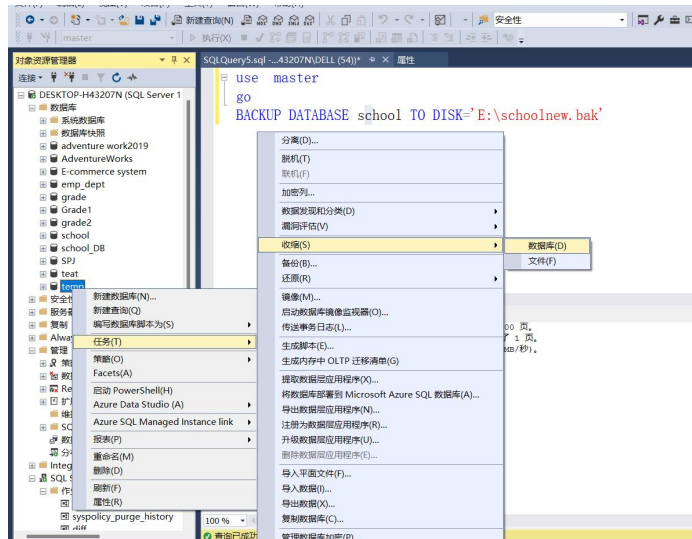


六 如何清除日志文件。

使用 SQL Server Management Studio (SSMS)：打开 SSMS，连接到目标 SQL Server 实例。右键单击数据库，选择“任务”->“备份”。在“备份类型”下拉菜单中选择“日志”。选择要备份的数据库，并为备份指定一个位置。然后点击“确定”执行备份操作。备份完成后，可以删除旧的日志文件。

使用 T-SQL 命令：打开 SQL Server Management Studio 或任何其他 SQL Server 查询工具，连接到目标 SQL Server 实例。

七 使用企业管理器练习数据库的压缩



选择页

常规

脚本

帮助

通过收缩所有数据库文件释放未使用的空间，可以减小数据库的大小。若要收缩单个数据库文件，请使用“收缩文件”。

数据库(D):

temp

数据库大小

当前分配的空间(C):

16.00 MB

可用空间(A):

4.23 MB (26%)

收缩操作

☐ 在释放未使用的空间前重新组织文件。选中此选项可能会影响性能(R)。

收缩后文件中的最大可用空间(M):

0 %

连接

服务器: DESKTOP-H43207N

连接: DESKTOP-H43207N\DESKTOP-H43207N\SQL Server Enterprise Manager

八 把 school 备份到其他计算机上。

复制数据库向导

选择传输方法

您要如何传输数据?

☒ 使用分离和附加方法(D)

此方法速度较快，但要求源数据库脱机。在升级数据库或移动非常大的数据库时，此方法最佳。使用此选项时，与源数据库之间不能有任何用户连接。

☒ 如果失败，则重新附加源数据库(R)

☐ 使用 SQL 管理对象方法(M)

此方法速度较慢，但源数据库可以一直处于联机状态。如果数据选择此项可使用 SQL 管理对象的表，复制操作可能会失败。



[加上 如果没有 norecover ,发生的情况]
[企业管理期恢复 如何实现 norecover]

整理实验 16-18 为实验报告“实践综合实验五：事务”，提交到 csteaching 平台“实践综合实验五：事务”，截止日期：6.15.