

# 上机实验报告

课程名: \_\_\_\_\_

指导教师: \_\_\_\_\_

姓名	班级	实验得分
	学号	
实验时间	实验地点	
实验主题	数据库的备份与恢复	
实验目的	<p>(1) 通过实验, 深入理解故障的种类(特别是事务故障与系统故障), 深入理解故障产生的原因及所带来的数据问题。</p> <p>(2) 通过实验, 深入理解: 数据备份的作用与意义, 数据备份的分类(动态转储/静态转储、海量转储/增量转储), 日志文件的作用、格式、内容、用途、登记逻辑, 深入理解数据库各种备份策略及策略的灵活应用。</p> <p>(3) 通过实验, 深入理解: 数据库故障和恢复策略(事务故障及其恢复、系统故障及其恢复、介质故障及其恢复), 检查点设置的机制与逻辑, 数据恢复模式定制, 后备副本与日志文件的结合逻辑及应用。</p> <p>(4) 通过实验, 能将 SQL Server 恢复子系统功能应用与 SQL 编程相结合, 制定科学的数据库备份与恢复策略, 并有效实现数据的备份与恢复。</p>	
实验内容	<p>(1) 故障恢复</p> <p>a、事务故障恢复: 编写 SQL 代码: 构建一个事务 T1, 对数据库进行操作。在完成对数据部分提交(保存到磁盘的物理数据库中)后, 此时发生事务故障(代码中不执行回滚)。现要求利用日志文件, 实现对数据库的恢复, 使数据库处于一致性状态。</p> <p>事务故障恢复逻辑的提示: ①反向扫描日志文件, 查找该事务的更新操作。②对该事务的更新操作执行反操作, 即对已经插入的新记录进行删除操作, 对已删除的记录进行插入操作, 对修改的数据恢复旧值(用旧值代替新值)。这样由后向前逐个扫描该事务已做所有更新操作, 并做同样处理, 直到扫描到此事务的开始标记, 事务故障恢复完毕。</p> <p>b、系统故障恢复: 编写 SQL 代码: 构建两个事务 T1 和 T2, 同时都对数据库进行操作。事务 T1 尚未执行完成, 但对数据库的更新已写入数据库; 事务 T2 已执行完成, 但数据尚在缓存区, 没有提交。此时发生系统故障。现要求利用日志文件, 实现对数据库的恢复, 使数据库处于一致性状态。</p> <p>系统故障恢复提示: ①正向扫描日志文件: (a) 查找尚未完成的事务, 将</p>	

其事务标识记入撤销队列。(b) 同时查找已提交的事务，将其事务标识记入重做队列。②对撤销队列中的各个事务进行撤销处理。③对重做队列中的各个事务进行重做处理。

(2) 数据库备份 a、编写 SQL 代码，对用户数据库 STUDY 进行完整备份。b、编写 SQL 代码，定期对数据库 STUDY 进行完整备份（如在每天凌晨 2 点开始数据库的全备份）。c、编写 SQL 代码，定期对数据库进行完整差异备份（如：在每月 1 日的凌晨 2 点，对数据库进行完整备份；每月其他时间的凌晨 2 点，对数据库进行差异/增量备份）。【假设】数据库在 Ta 时刻，停止所有事务。直到备份完成后，才继续运行事务。

(3) 数据库恢复 a、假设：数据库在 Ta 时刻，进行了数据库的完整备份，形成了一份副本。在 Ta 时刻之后，运行若干事务。此时数据库发生故障。编写 SQL 代码：利用数据库完整副本，并结合日志文件，对数据库进行恢复。b、假设：数据库在 Ta 时刻，有一份数据完整备份副本。在 Ta 时刻之后，运行若干事务，并数据库进行了差异备份。产生差异备份后，又运行了若干事务。此时此时数据库发生故障。编写 SQL 代码：利用数据库的完整差异副本，并结合日志文件，对数据库进行恢复。

实验  
结果  
/  
实验  
结论

### (1) 故障恢复

在事务过程中发生错误后，可由两条命令查看事务日志

DESKTOP-0B31L2E5M - dbo.C SQLQuery1.sql - lo...0B31L2E5M (SZ)\*

```
BEGIN TRAN TEST123
INSERT INTO C VALUES('C01022','TEST',NULL,1);
INSERT INTO C VALUES('C01023','TEST',NULL,1);
select * from fn_dblog(null,null)
DBCC log (SM)
```

结果 消息

1	00000018:0000010c:0001	LOP_BEGIN_CKPT	LCX_NULL	0000:00000000	0	0x0000	96	96	0000
2	00000018:0000010d:0001	LOP_END_CKPT	LCX_NULL	0000:00000000	0	0x0000	136	136	0000
3	00000018:0000010e:0001	LOP_BEGIN_XACT	LCX_NULL	0000:0000032a	0	0x0000	64	92	0000
4	00000018:0000010e:0002	LOP_MODIFY_ROW	LCX_BOOT_PAGE	0000:0000032a	0	0x0000	62	80	0000
5	00000018:0000010f:0001	LOP_PREP_XACT	LCX_NULL	0000:0000032a	0	0x0000	64	68	0000
6	00000018:00000110:0001	LOP_COMMIT_XACT	LCX_NULL	0000:0000032a	0	0x0000	48	52	0000

Current LSN	Operation	Context	Transaction ID	LogBlockGeneration
00000018:0000010c:0001	LOP_BEGIN_CKPT	LCX_NULL	0000:00000000	0
00000018:0000010d:0001	LOP_END_CKPT	LCX_NULL	0000:00000000	0
00000018:0000010e:0001	LOP_BEGIN_XACT	LCX_NULL	0000:0000032a	0
00000018:0000010e:0002	LOP_MODIFY_ROW	LCX_BOOT_PAGE	0000:0000032a	0
00000018:0000010f:0001	LOP_PREP_XACT	LCX_NULL	0000:0000032a	0
00000018:00000110:0001	LOP_COMMIT_XACT	LCX_NULL	0000:0000032a	0

数据库的恢复由系统自动完成，无需人手动操作

### (2) 系统故障恢复

正向扫描日志文件：(a) 查找尚未完成的事务，将其事务标识记入撤销队列。(b) 同时查找已提交的事务，将其事务标识记入重做队列。对撤销队列中的各个事务进行撤销处理。对重做队列中的各个事务进行重做处理。

该步骤也是由 mssql 重启时自动完成，用户无需操作。

### (3) 数据库备份与恢复

对 study 进行完整备份

```
backup database SM to disk = 'C:\Users\zcy\Desktop\tmp\1.db'  
with noinit;  
restore database SM FROM disk = 'C:\Users\zcy\Desktop\tmp\1.db'  
with RECOVERY  
USE SM1;  
SELECT *FROM S;
```

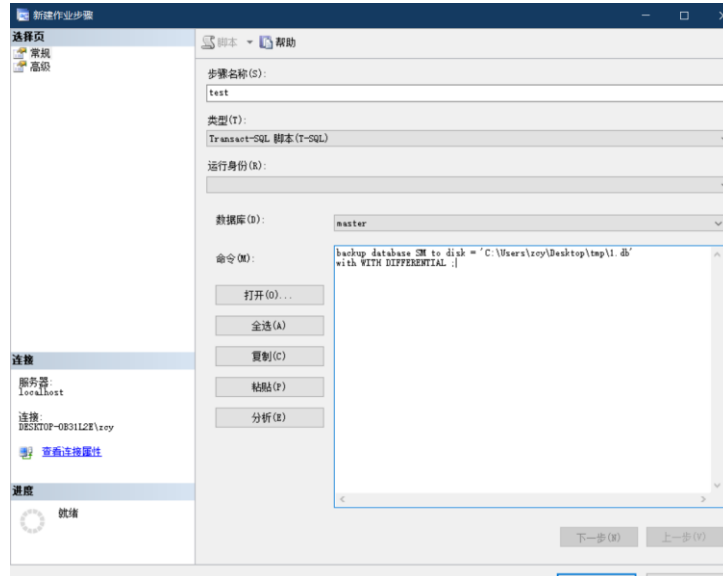
#### 消息

已为数据库 'SM'，文件 'SM' (位于文件 1 上)处理了 240 页。  
已为数据库 'SM'，文件 'SM\_log' (位于文件 1 上)处理了 1 页。  
BACKUP DATABASE 成功处理了 241 页，花费 0.018 秒(104.600 MB/秒)。

编写 SQL 代码，定期对数据库 STUDY 进行完整备份



## 编写 SQL 代码，定期对数据库进行完整差异备份



## 还原 a

```
use master
declare @dbname varchar(20)
set @dbname='SM'

declare @sql nvarchar(500)
declare @spid int--SPID 值是当用户进行连接时指派给该连接的一个唯一的整数
set @sql='declare getspid cursor for
select spid from sysprocesses where dbid=db_id(''+@dbname+'')'
exec (@sql)
open getspid
fetch next from getspid into @spid
while @@fetch_status<>-1--如果FETCH 语句没有执行失败或此行不在结果集中。
begin
exec('kill '+@spid)--终止正常连接
fetch next from getspid into @spid
end
close getspid
deallocate getspid
restore database SM FROM disk = 'C:\Users\zey\Desktop\tmp\1.db'
with REPLACE
```

消息  
2 为数据库 'SM', 文件 'SM' (位于文件 1 上)处理了 240 页。  
3 为数据库 'SM', 文件 'SM\_log' (位于文件 1 上)处理了 1 页。  
RESTORE DATABASE 成功处理了 241 页, 花费 0.007 秒(268.973 MB/秒)。

## 还原 b

### 差异备份及其还原

```
use master
backup database SM to disk = 'C:\Users\zey\Desktop\tmp\1.db'
with DIFFERENTIAL;
declare @dbname varchar(20)
set @dbname='SM'

declare @sql nvarchar(500)
declare @spid int--SPID 值是当用户进行连接时指派给该连接的一个唯一的整数
set @sql='declare getspid cursor for
select spid from sysprocesses where dbid=db_id(''+@dbname+'')'
exec (@sql)
open getspid
fetch next from getspid into @spid
while @@fetch_status<>-1--如果FETCH 语句没有执行失败或此行不在结果集中。
begin
exec('kill '+@spid)--终止正常连接
fetch next from getspid into @spid
end
close getspid
deallocate getspid
restore database SM FROM disk = 'C:\Users\zey\Desktop\tmp\1.db'
with REPLACE
```

消息  
2 为数据库 'SM', 文件 'SM' (位于文件 1 上)处理了 240 页。  
3 为数据库 'SM', 文件 'SM\_log' (位于文件 1 上)处理了 1 页。  
RESTORE DATABASE 成功处理了 241 页, 花费 0.007 秒(268.973 MB/秒)。

---

实验心得	通过这次实验，我学会了数据库的备份与恢复，知道了数据库异常的种类和解决办法。在实验过程中，我遇到了进程冲突和权限不够的问题。通过查阅资料，编写杀死相关进程的脚本和修改文件夹权限来完成任务。
------	--