# 实验3

# 主题14、视图

目的：掌握试图的建立、使用

任务1：建立视图

（1）建立学生学号、姓名、性别、课程号、成绩的视图 V\_sc，查看V\_sc中的数据。

（2）建立学生学号、姓名、出生年份的视图 V\_age，查看V\_age中的数据。

（3）建立 ‘JSJ’ 系的学生学号、姓名、性别、年龄的视图 V\_jsj。

（4）建立每门课程的平均分的视图 V\_avggrade。

（5）将视图 V\_jsj 中某学生（指定姓名，如张三）的年龄改为21岁。

（6）察看 student 中该学生（题（5）中学生，即张三）的年龄，查看 V\_age 中该学生的出生年月。

（7）查询每门课程的及格率。

思考：

* 利用 V\_jsj视图，可以更新SX系的学生的年龄吗? 写出理由。

如: update V\_jsj set sage=25 where sno=’ 0004’，0004 号学生为 SX 系.

* create view V\_student (sno, sname, ssex, sage, sdept) as

select sno, sname, ssex, sage, sdept from student

学籍管理系统使用此视图。现把表student 的列名 sname 改为 xm，请问原来的学籍管理系统能正常运行吗？如不能，则如何处理能使系统正常运行。

# 主题15、安全性控制

目的：掌握SQL Server授权机制

任务1：访问控制

（1）建立新用户Mary , 密码1234。

（2）授予 Mary 可以访问 School 数据库的权力。

（3）以Mary 登录 SQL Server ,

执行select \* from student, 记录执行结果，说明原因。

（4）将 course 的查询、更改权限授予 Mary。

（5）把查询 student 表和修改学生学号的权限授予用户 Mary，且能转授他人。

（6）把对 course 表的更改权限从Mary 收回。

（7）把第（5）小题授予Mary的权限收回。

（8）Mary 只能查询 ‘1001’ 号课程的学生成绩，请问如何授权。

思考:

* sp\_addlogin , sp\_grantdbaccess 语句的区别
* 如有200个人需要授权，SQL Server如何简化授权机制。

# 主题16、存储过程

目的: 掌握存储过程的概念、编程及使用

任务1：编写以下存储过程

1. 编写一个存储过程 usp\_avgage , 向客户端返回每个系科的学生平均年龄。

系科 平均年龄

JSJ 21

SX 20

…….

1) 编写存储过程的代码

2）调试、运行该存储过程。

1. 编写一个存储过程 usp\_sdept, 传入一个系科代码，返回该系的平均年龄，人数。
2. 编写存储过程 usp\_updateGrade , 传入参数为课程号,处理逻辑:

对传入的这门课,进行如下处理:

如某学生该门课成绩>80 , 则加 2 分

如某学生该门课成绩>60 , 则加 1 分

如某学生该门课成绩<=60 ,则减 1分

并且返回此门课的每个学生的最新成绩: 学号 成绩.

1. 编写存储过程 usp\_comp\_age , 比较0001，0002学生的年龄的高低，输出： XXXX学生的年龄大。注意：XXXX为学生的姓名
2. 编写存储过程 usp\_comp\_age1 , 比较两个学生的年龄的高低，两个学生的学号有参数输入，最后输出： XXXX学生的年龄大。注意： XXXX为学生的姓名。
3. 编写存储过程 usp\_comp\_age2 , 比较两个学生的年龄的高低，两个学生的学号有参数输入，最后把年龄大的学生的姓名、性别返回客户端。
4. 编写存储过程 usp\_t1，传入参数为学号,把该学号的课程1001的成绩减到58分。每次只能减1分，用循环完成。
5. 编写存储过程 usp\_disp , 传入参数为课程号,处理逻辑:

返回每个学生的成绩等级：成绩>=90为优,成绩>=80为良,成绩>=70为中,成绩>=60为及格,成绩<=60为不及格。

返回结果如下：

学号 课程号 成绩 等第

0001 1001 91 优

0001 1002 78 中

……………..

1. 编写一个存储过程，传入参数为 系科，执行后，把该系科的学生按如下格式输出学生成绩：

学号 姓名 1001 课程 1002课程 1003 课程 平均分

1. 编写存储过程，统计男女生1001，1002，1003各自的选修人数，输出格式如下：

性别 1001人数 1002人数 1003人数 小计

男 3 5 2 10

女 2 4 1 7

合计 5 9 3 17

（数据为示意数据）

1. 编写一个存储过程，利用存储过程的参数返回数据库服务器上的日期时间。

思考：何时需要存储过程？

# 主题17、触发器

目的: 掌握触发器的概念、编程及使用

任务1: 建立学生表的触发器 usp\_addstudent，当增加学生时，SX系的学生不能超过30岁。

1. 写出触发器
2. 执行下列语句块，并观察该学生是否加入到 student。

begin tran

insert into student (sno,sname,ssex,sage,sdept)

values (‘0701’,’刘欢’,’男’,26,’SX’)

if @@error=0

commit

else

rollback

1. 执行下列语句块，并观察该学生是否加入到 student。

begin tran

insert into student (sno,sname,ssex,sage,sdept) values (‘0702’,’赵欢’,’男’,31,’SX’)

if @@error=0

commit

else

rollback

任务2：实现下列触发器，并做相应测试。

1. 不能删除年龄大于25岁的学生记录。
2. 建立触发器 usp\_delcourse , 使课程表中1001，1002，1003 三门课不会被删除。
3. 对学生表建立一触发器，使更改后的年龄只能比原值大。
4. 对sc表建立触发器，使‘JSJ’系的学生不可选择 ‘1004’号课程
5. 对表 course 建触发器，实现级联删除的功能，但某课选修人数大于3则不能删除。

(先删除 sc 表对course 的外码)

思考：存储过程和触发器有何联系或异同？

# 主题18、恢复技术

目的：掌握数据库的备份及恢复的方法；了解备份方案的设定。

注意：关键步骤请截图

任务1：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’

1. 查看备份文件中的信息

RESTORE FILELISTONLY FROM DISK=’c:\schooldata.bak’

RESTORE HEADERONLY FROM DISK=’c:\schooldata.bak’

1. 恢复完全备份
2. 先删除数据库 school

USE Master

GO

DROP DATABASE school

1. 然后恢复.

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

1. 查看 school 的student 中的数据

任务2：建立差异备份

1. 建立备份
2. 制作数据文件备份 schoolDiff.bak
3. 把学号 7001， 姓名：王海，性别：男，年龄为23 的学生加入student
4. 制作school 的差异备份 ，存入schoolDiff.bak
5. 把学号 7002， 姓名：赵燕，性别：女,年龄为22 的学生加入student
6. 制作school 的差异备份 ，存入schoolDiff.bak
7. 查看备份文件 schoolDiff.bak 中的信息
8. 删除 school 数据库
9. 恢复数据库 school 到第(2)步状态，并使用select \* from student观察student 数据
10. 恢复数据库 school 到最新状态，并使用select \* from student观察student 数据

任务3：利用日志备份

1. 设置故障还原模型为：完全
2. 建立备份
3. 制作数据文件备份 schooldata1.bak
4. 把学号 7003， 姓名：王江，性别：男，年龄为23 的学生加入student
5. 制作日志备份存入 schoollog.bak
6. 把学号 7004， 姓名：赵兰，性别：女,年龄为22 的学生加入student
7. 制作日志备份存入 schoollog.bak
8. 观察schoollog中的信息
9. 删除 school 数据库
10. 利用schooldata1.bak 及 schoollog.bak 恢复数据库 school 到最新状态

任务4：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。
3. 手工启动两个备份作业
4. 删除 school 数据库
5. 利用1，2步的备份进行 school的恢复。

任务5：其他

1. 使用企业管理器练习数据库的分离及附加。
2. 清除日志文件。
3. 使用企业管理器练习数据库的压缩。
4. 把school 备份到其他计算机上。

# 主题19、事务

目的：掌握并理解事务

任务1、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’

1. 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’

此事务结束了吗？

1. 执行：

Select \* from student where sno=’0001’

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?

1. 执行下列语句。

ROLLBACK TRANsaction

然后再执行：

Select \* from student where sno=’0001’

观察0001的年龄, 解释发生这种现象的原因。

任务2、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’

1. 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’

1. 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?

任务3、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？

# 主题20、锁

目的：理解锁的概念及锁的作用

任务1、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

1. 设置锁定超时时间

SET LOCK\_TIMEOUT 5000

1. SP\_LOCK

任务2、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’

1. 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’

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）

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）

4) 强行终止上一步的命令，然后执行语句：

DBCC opentran

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。

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

任务3、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

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

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。

1. 执行下列语句

BEGIN TRAN

Update student set sage=sage + 1 where sno=’1001’

Print ‘server process ID (spid) : ‘

Print @@spid

1) 然后执行下列语句

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。

2) 然后执行下列语句

commit tran

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。

1. 使用 SET TRANSACTION ISOLATION LEVEL Serializable

然后重新执行前两步，观察与原来有何不同。

1. 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。

1. 了解锁定超时
2. 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’

1. 打开第二个连接

执行：

select \* from student

记录观察到的现象。

1. 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。

# 主题21、编程实例并撰写报告3

任务1、编写存储过程 usp\_update1, 传入参数为课程号，处理逻辑：

对传入的这门课,进行如下处理:

如某学生该门课成绩>80，则加 2 分

如某学生该门课成绩>60，则加 1 分

如某学生该门课成绩<=60，扣1分

要求：在存储过程中，要么全部学生的成绩被处理成功，要么全部不处理

思考：在调试中，采用那些措施，使存储过程运行时执行回滚操作 (rollback) 。

任务2、编写触发器，对insert、update语句进行监控，当学生的年龄超过40岁时，把该学生的系科改为’BAK’，同时删除该学生的所有选课记录。（注意，利用事务，使修改系科及删除选课记录要么全做，要么全不做）

任务3、撰写实验报告3

按实践要求和报告模板撰写实验报告3，本次实践内容包括主题14~21，并在指定时间前（届时通知）提交至csteaching实验管理平台，逾期不候！注意：**实验报告中请勿忘记填写学号姓名**，并以**学号姓名作为该Word文件名，如12345678张三-实验报告3**。