目录

[数据库实验一 1](#_Toc137736795)

[数据库实验二 2](#_Toc137736796)

[数据库实验三 6](#_Toc137736797)

[数据库实验四 10](#_Toc137736798)

[数据库实验五 15](#_Toc137736799)

[数据库实验六 20](#_Toc137736800)

数据库实验一

1. 用命令方式删除数据库jxsk（附上T-SQL语句）

USE master  
GO  
DROP DATABASE jxsk202030071077

1. 命令方式创建数据库：数据库名称为StudentDB(学生数据库)。数据库包含一个主要数据文件，一个次要数据文件，和一个日志文件。数据库参数自拟。 (附上T-SQL语句)

CREATE DATABASE StudentDB  
    ON PRIMARY  
    (  
        NAME= 'StudentDB\_data',  
        FILENAME='D:\zz\_mydata\StudentDB\_data.mdf',  
        SIZE = 10MB  
    ),  
    (  
        NAME = 'StudentDB\_data2',  
        FILENAME='D:\zz\_mydata\StudentDB\_data2.ndf'  
    )  
        LOG ON  
    (  
        NAME='StudentDB\_log',  
        FILENAME='D:\zz\_mydata\StudentDB\_data.ldf'  
    )

1. 命令方式修改数据库：(附上T-SQL语句)
   1. 增加一个次要文件
   2. 修改主要数据文件的增长方式
   3. 删除一个次要文件

ALTER DATABASE StudentDB  
ADD FILE  
(  
    NAME='StudentDB\_data3',  
    FILENAME='D:\zz\_mydata\StudentDB\_data3.ndf'  
)  
ALTER DATABASE StudentDB  
MODIFY FILE  
(  
    NAME='StudentDB\_data',  
    FILEGROWTH=5%  
)  
ALTER DATABASE StudentDB  
REMOVE FILE StudentDB\_data2

插入数据的SQL语句格式类似如下语句。

insert into SC

values ('S3','C1',87),

('S3','C3',80),

('S3','C5',87)

数据库实验二

1. 向S表中增加可以date类型的列，不允许为空。测试是否能成功，分析其原因。

alter table S add dateday date not null

ALTER TABLE 只允许添加满足下述条件的列: 列可以包含 Null 值；或者列具有指定的 DEFAULT 定义；或者要添加的列是标识列或时间戳列；或者，如果前几个条件均未满足，则表必须为空以允许添加此列。

1. 将dept列修改为Unicode字符类型nchar(4)的列，测试是否能成功。

“Teacher”表

- 警告: 从“nchar(10)”转换列“dept”时可能会丢失数据。

1. 向C表中插入数据(C8，软件工程，NULL，3，NULL，T3)。测试是否成功，分析其原因

insert into Course(cno,cname,hours,credit,pno,tno)

values('C8','软件工程',NULL,3, NULL,'T3')

消息 515，级别 16，状态 2，第 1 行

不能将值 NULL 插入列 'hours'，表 'StudentDB.dbo.Course'；列不允许有 Null 值。INSERT 失败。

语句已终止。

1. 向C表中插入数据(C9，操作系统，48，NULL，NULL，T5)。测试是否成功，分析其原因

insert into Course(cno,cname,hours,credit,pno,tno)

values('C9','操作系统',48,NULL,NULL,'T5')

消息 515，级别 16，状态 2，第 1 行

不能将值 NULL 插入列 'credit'，表 'StudentDB.dbo.Course'；列不允许有 Null 值。INSERT 失败。

语句已终止。

1. 用命令实现以下操作：

在删除操作之前，请对数据库备份。

1. 用SQL命令再创建一个学生表Stu，结构同S表

create table Stu

(

sno char(10) not null primary key,

sname char(8) not null ,

age tinyint not null default(18) check(age>=0and age<=120),

sex char(2) not null default('男') check(sex='男'or sex='女'),

birthdate date null ,

dept nchar(10) not null check(dept='计算机'or dept='人工智能'or dept='软件工程'or dept='物联网'),

check (datediff(year,getdate(),birthdate)=age)

)

命令已成功完成。

1. 用SQL命令删除学生表Stu

drop table Stu;

命令已成功完成。

1. 把学生“周武”的年龄改为20，系别改为“人工智能”

update Student set dept='人工智能'

where sname='周武';

(1 行受影响)

1. 将“张小明”同学的信息从基本表S中删除

delete from Student

where sname = '张小明'

(1 行受影响)

1. 将所有学生的年龄+1

Update Student set age=age+1

1. 将所有不及格学生的选课记录删除

delete from SC

where score <60

(0 行受影响)

数据查询

1. 查询计算机系的所有教师

select \* from Teacher

where dept = '计算机';

1. 查询年龄在18~20岁（包括18和20）之间的所有学生的信息

select \* from Student

where age >= 18 and age <=20;

1. 查询年龄小于20岁的所有男同学的学号和姓名

select sno,sname from Student

where sex = '男' and age <=20;

1. 查询所有同学的学号、姓名和出生年份

select sno,sname,birthdate from Student;

1. 查询姓“李”的所有学生的姓名、年龄和性别

select sname,age,sex from Student

where sname like '李% '

1. 查询所有女同学的选课情况

select sname,Course.cno,cname,hours,credit,pno,tno

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where Student.sex = '女';

1. 查询没有考试的同学选课情况

select \* from SC

where score is null

1. 查询计算机系教师开设的所有课程的课程号和课程名

select cno,cname

from Course join Teacher on Course.tno = Teacher.tno

where Teacher.dept = '计算机'

1. 查询至少有一门成绩高于90分的学生姓名和年龄

select sname,age

from Student join SC on Student.sno = SC.sno

group by Student.sname,age

having max(SC.score) >90;

1. 查询选修“微机原理”的所有学生的姓名和成绩

select sname ,score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where Course.cname = '微机原理'

1. 列出学生选课情况，包括课程名、学生名、是否及格（>=60及格，<60不及格，NULL未考试）

select cname ,sname ,score,

case score

when score >=60 then '及格'

when score <60 then '不及格'

when score = null then '未考试'

from Student join SC on Student.sno=SC.sno

join Course on Course.cno=SC.cno

1. 查询女同学的人数

select count(\*) as 女同学人数 from Student

where sex = '女'

1. 查询男同学的平均年龄

select avg(age) as 平均年龄 from Student

where sex = '男'

1. 查询男、女同学各有多少人

select Student.sex ,count(sex) as 人数 from Student

group by sex;

1. 查询每门课的选课人数，没有人选的不要列出

select SC.cno,cname,count(\*) as 选课人数

from SC join Course on SC.cno = Course.cno

group by SC.cno,cname

1. 查询每个同学的选课门数，没有选课的同学也要列出

select sname,SC.sno,count(\*) as 选课门数

from SC join Student on SC.sno = Student.sno

group by SC.sno,sname

1. 查询选修了和张小明同学同一门课的学生姓名和课程名。

select sname , cname from Student join SC on Student.sno=SC.sno

join Course on SC.cno=Course.cno

where sname!= '张小明'and SC.cno in

(select cno from SC where sno in

(select sno from Student where sname='张小明') )

数据库实验三

1. 试算所有“数据结构”成绩60分以下的统一增加10分后，仍有多少人不及格。

update SC

set score = score +10

where cno = 'C4' and score <60;

select count (\*) as '不及格人数'

from SC join Course on SC.cno = Course.cno

where score <60 and cname = '数据结构'

1. 查询每门课程的学生选修人数(只输出超过10人的课程),要求输出课程号和课程名及选修人数,查询结果按人数降序排列,若人数相同,按课程号升序排列

select Course.cno,cname,count(distinct sno)as num

from Course join SC on Course.cno = SC.cno

group by Course.cno ,cname

having count(distinct sno) >10

order by num desc,cno asc

1. 查询每门课程的学生选修人数(所有课程都要输出),要求输出课程号和课程名及选修人数,查询结果按人数降序排列,若人数相同,按课程号升序排列

select Course.cno,cname,count(distinct sno)as num

from Course join SC on Course.cno = SC.cno

group by Course.cno ,cname

order by num desc,cno asc

1. 查询至少选修了两门课程的学生学号和姓名

select Student.sno,sname

from Student join SC on Student.sno = SC.sno

group by Student.sno,sname

having count(cno) >=2

1. 查询选修了“计算机”系教师所讲授的课程的学生姓名和成绩

select sname ,cname,score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where Teacher.dept = '计算机'

1. 统计各科课程号、课程名、选课人数、平均分、最高分、最低分，并存入新表statscore

select SC.cno,cname,count(distinct sno) as '选课人数',

avg(score) as'平均分',max(score) as'最高分',min(score) as'最低分'

into statscore

from SC join Course on SC.cno = Course.cno

group by SC.cno,cname

**以下操作采用无关子查询实现**

1. 查询至少选修课程号为“c4”和“c6”两门课程的学生的学号,姓名

select sno,sname from Student

where sno in

(select sno from SC where cno = 'C6' and cno = 'C4')

1. 查询选修了“高等数学”或“普通物理”的学生姓名

select sname from Student

where sno in

(select sno

from SC join Course on SC.cno = Course.cno

where cname = '高等数学' or cname = '普通物理')

1. 查询未选修王老师所讲授任意课程的学生的学号和成绩

select Student.sno,score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where Student.sno not in (select Student.sno

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where tname = '王老师' )

1. 查询选修了“计算机”系教师所讲授的课程的学生姓名和成绩

select sname,score

from Student join SC on Student.sno = SC.sno

where SC.sno in (select SC.sno

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where Teacher.dept = '计算机'

)

select sname,Course.cname,score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where SC.sno in (select SC.sno

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where Teacher.dept = '计算机'

)

1. 查询学号比“张小明”同学大而年龄比他小的学生姓名

select sname from Student

where sno in (select sno from Student

where sno > (select sno from Student where sname = '张小明')

and age < (select age from Student where sname = '张小明')

)

1. 查询年龄大于所有女同学年龄的男学生的姓名和年龄

select sname,age from Student

where sno in (select sno from Student

where birthdate < (select min(birthdate) from Student where sex = '女')

)

1. 查询至少选修了两门课程的学生学号和姓名

select sno,sname from Student

where sno in(select sno from SC

group by SC.sno having count(SC.cno)>=2 )

1. 查询未选修“高等数学”的学生的学号和姓名

select sno,sname from Student

where sno not in(select sno from SC join Course on SC.cno = Course.cno

where cname = '高等数学' )

1. 查询不是计算机系教师所讲授的课程的课程名和课程号

select cno,cname from Course

where Course.cno not in(select Course.cno

from SC join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where Teacher.dept = '计算机' )

1. 查询哪些学生所选的课程是由本院系的教师教的，列举学生姓名、课程名和教师名

select sname,cname,tname

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where Student.sno in(select Student.sno

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

join Teacher on Course.tno = Teacher.tno

where Teacher.dept = Student.dept )

1. 查询年龄大于女同学平均年龄的男学生的姓名和年龄

select sname,age from Student

where sno in (select sno from Student

where age > (select avg(age) from Student where sex = '女')

and sex = '男'

)

1. 查询至少讲授两门课程的教师姓名和其所在系

select tname,dept from Teacher

where Teacher.tno in(select Teacher.tno from Teacher

join Course on Teacher.tno = Course.tno

group by Teacher.tno having count(Course.cno)>=2 )

数据库实验四

**以下操作采用相关子查询完成：**

1. 查询选修了“高等数学”或“普通物理”的学生姓名

select sname

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where exists(select \*

where cname = '高等数学' or cname = '普通物理' and

exists (select \* from SC where Student.sno = SC.sno and SC.cno = Course.cno))

1. 查询未选修“高等数学”的学生的学号和姓名

select distinct Student.sno,sname from Student

where Student.sno not in (select sno from SC where cno = (select cno from Course where cname = '高等数学'))

1. 查询未选修王老师所讲授任意课程的学生的学号和成绩

select Student.sno,score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where not exists (select \* from Course join Teacher on Course.tno = Teacher.tno

where tname like '王%' and exists

(select \* from SC where Student.sno = SC.sno and SC.cno = Course.cno))

1. 查询选修了“计算机”系教师所讲授的课程的学生姓名和成绩

select Student.sname,score

from Student join SC on Student.sno = SC.sno

where exists (select \* from Course

where Course.cno = SC.cno and exists

(select \* from Teacher where Teacher.tno = Course.tno and Teacher.dept='计算机'))

1. 查询选修了王平老师所讲授所有课程的学生的学号和成绩

select Student.sno,score

from Student join SC on Student.sno=SC.sno

join Course on Course.cno=SC.cno

join Teacher on Teacher.tno=Course.tno

where not exists

(

select \*

from Course join Teacher on Course.tno=Teacher.tno

where tname='王平' and not exists

(

select \*

from SC

where Course.cno=SC.cno and sc.sno=Student.sno

)

)and tname='王平'

1. 查询被全部女生选修了的课程的课程号和课程名

select cno,cname

from Course

where exists (select \* from Student join SC on Student.sno = SC.sno

where Course.cno = SC.cno and Student.sex = '女')

**视图操作：**

1. 创建视图view\_s，包含学号，姓名，性别和系别。

go

create view view\_s

as

select sno,sname,sex,dept

from Student

go

1. 创建视图view\_cj，包含学号，姓名，课程名和成绩。

go

create view view\_cj

as

select Student.sno,sname,cname,score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

go

1. 从view\_s基础上创建view\_ms，包含所有男生的学号，姓名和院系。

go

create view view\_ms

as

select sno,sname,dept

from view\_s

where sex = '男'

go

1. 创建视图view\_max\_min(cno,MAX,MIN)，包含所有课程的课程号，最高成绩和最低成绩。

go

create view view\_max\_min(cno,MAX,MIN)

as

select cno,max(score),min(score)

from SC

group by cno

go

1. 创建视图V\_FAIL(sname,cname,score)，包含成绩不及格的学生名，课程名和成绩

go

create view V\_FAIL(sname,cname,score)

as

select sname,cname,score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where score <60

go

1. 将视图view\_s的性别列删除，并且增加年龄列，且只能看到年龄18岁及以下的学生，设置确保用户基于视图的修改不会导致记录脱离本视图。

go

alter view view\_s

as

select sno,sname,age,dept

from Student

where age<=18

go

1. 基于视图view\_s插入数据(S100,韩磊,40,计算机系)，观察是否能成功，不成功分析原因。

insert into view\_s

values ('S100','韩磊',40,'计算机系')

(1 行受影响)

成功，因为虽然只显示18岁以下的学生，但是超过18岁的学生依然可以被插入，只不过被隐藏了

1. 基于视图view\_s插入数据(S101,韩磊,17,计算机系)，观察是否能成功，不成功分析原因。

insert into view\_s

values ('S101','韩磊',17,'计算机系')

(1 行受影响)

成功

1. 通过视图view\_cj修改某学生的成绩。

update view\_cj

set score = score + 2

where sname = '李思'and cname = '高等数学'

(1 行受影响)

1. 基于视图view\_cj插入一条数据，观察是否能成功，不成功分析原因。

insert into view\_cj

values ('S19','韩磊','普通物理',84)

消息 4405，级别 16，状态 1，第 1 行

视图或函数 'view\_cj' 不可更新，因为修改会影响多个基表。

不成功，因为修改会影响多个基表。

1. 基于视图view\_cj删除一条数据，观察是否能成功，不成功分析原因。

delete from view\_cj where sname = '李思'and cname = '计算机基础'

消息 4405，级别 16，状态 1，第 1 行

视图或函数 'view\_cj' 不可更新，因为修改会影响多个基表。

不成功，因为修改会影响多个基表。

1. 删除视图view\_s，观察是否能成功，不成功分析原因。

drop view view\_s

命令已成功完成。

1. 是否能往视图view\_max\_min(cno,MAX,MIN)中增加、修改或者删除数据。

insert into view\_max\_min

values ('C7',98,84)

消息 4406，级别 16，状态 1，第 1 行

对视图或函数 'view\_max\_min' 的更新或插入失败，因其包含派生域或常量域。

update view\_max\_min

set MIN = MIN + 2

where cno = 'C1'

消息 4406，级别 16，状态 1，第 1 行

对视图或函数 'view\_max\_min' 的更新或插入失败，因其包含派生域或常量域。

delete from view\_max\_min where cno = 'C1'

消息 4403，级别 16，状态 1，第 1 行

因为视图或函数 'view\_max\_min' 包含聚合、DISTINCT 或 GROUP BY 子句或者 PIVOT 或 UNPIVOT 运算符，所以无法进行更新。

**安全性控制：**

1. 创建登录名STU，密码1234，不设置服务器角色和用户映射，观察用新建登录名是否能访问数据库，并分析原因。（注意新建后重启）

create login STU with password='1234'

新建用户名无法访问数据库，因为没有用户与数据库之间的映射关系。

1. 启用StudentDB数据库中Guest账户，设置数据库角色为db\_datareader，观察用STU登录名是否能访问数据库，是否能查看数据，是否能修改数据，并分析原因。

use [StudentDB]

go

grant connect to guest

exec sp\_addrolemember 'db\_datareader','guest'

可以访问数据库，不可以修改数据。因为Guest的数据库角色为db\_datareader，它只有访问数据库的权限，不能修改。

1. 修改登录名STU，将其映射到StudentDB数据库上，观察StudentDB数据库中用户名的变化。

StudentDB数据库中用户名多了STU

1. 给用户名STU设置不同的数据库角色，测试其权限。
2. 为STU设置db\_datareader角色，可以查询数据但不能修改数据

exec sp\_addrolemember 'db\_datareader','STU'

1. 为STU设置db\_datawriter角色，不可以查询数据但可以修改数据  
   exec sp\_addrolemember 'db\_datawriter','STU'
2. 为STU设置db\_accessadmin角色

exec sp\_addrolemember 'db\_accessadmin','STU'

为登录名为STU2的账户在StudentDB中创建用户，映射名字为stu2,使得STU2能够访问数据库StudentDB  
当删除此角色后就不能执行了

1. 为STU设置db\_securityadmin角色

exec sp\_addrolemember 'db\_securityadmin','STU'

1. 为STU设置db\_denydatareader角色,不能对数据库中任何表执行SELECT操作

exec sp\_addrolemember 'db\_denydatareader','STU'

1. 为STU设置db\_denydatawriter角色,可以查看数据，但不能对数据库中任何表执行UPDATE、DELETE和INSERT操作

exec sp\_addrolemember 'db\_denydatawriter','STU'

数据库实验五

1. 创建4个登录名：D01(计算机系主任)，Dean01(院长)，STUManage（学生管理员），Dorm01(宿舍管理员)，分别设置授予以下权限：

|  |  |  |  |
| --- | --- | --- | --- |
| S | O | t | P |
| 宿舍管理员 | Student(sno,sname,sex,dept) | 读 |  |
| 学生管理员 | Student | 增删改查 |  |
| 计算机系主任 | Student | 读 | Dept=’计算机’ |
| 院长 | Student | 读 |  |

CREATE LOGIN Dorm01

WITH PASSWORD='123456'

CREATE USER Dorm01 FOR LOGIN Dorm01

Grant select(sno,sname,sex,dept) on Student to Dorm01

命令已成功完成。

CREATE LOGIN STUManage WITH PASSWORD='123456'

CREATE USER STUManage FOR LOGIN STUManage

Grant select,insert,update,delete on Student to STUManage

命令已成功完成。

CREATE LOGIN D01 WITH PASSWORD='123456'

CREATE USER D01 FOR LOGIN D01

命令已成功完成。

create view cs\_stu

as

select \*

from Student

where dept='计算机'

命令已成功完成。

Grant select on cs\_stu to D01

命令已成功完成。

CREATE LOGIN Dean01 WITH PASSWORD='123456'

CREATE USER Dean01 FOR LOGIN Dean01

Grant select on Student to Dean01

命令已成功完成。

1. 用SQL语句完成下列权限管理
2. 新建一个名字为”wang”的登录名，密码是”123456”；

create login wang with password = '123456'

命令已成功完成。

1. 把账号“wang”映射到数据库“StudentDB”上的STU\_wang用户；

use StudentDB

go

create user STU\_wang for login wang

go

exec sp\_change\_users\_login 'UPDATE\_ONE','STU\_wang','wang'

命令已成功完成。

1. 将登录名”wang”修改为“STU\_wang”;

alter login wang with name = STU\_wang

命令已成功完成。

1. 将登录名“STU\_wang”的密码改为111111；

alter login STU\_wang with password = '111111'

命令已成功完成。

1. 将登录名“STU\_wang”加入dbcreater服务器角色中。

exec sp\_addsrvrolemember 'STU\_wang','dbcreator'

命令已成功完成。

1. 将用户名“STU\_wang”加入到db\_datareader数据库角色中。

exec sp\_addrolemember 'db\_datareader','STU\_wang'

命令已成功完成。

1. 授予STU\_wang对STUDENT表格的所有权限

grant all privileges on Student to STU\_wang

ALL 权限已不再推荐使用，并且只保留用于兼容性目的。它并不表示对实体定义了 ALL 权限。

1. 收回STU\_wang对STUDENT表格的读权限，测试是否能查看STUDENT表格；

revoke select on Student from STU\_wang

命令已成功完成。

select \* from Student

可以查看Student表格

1. Deny STU\_wang对STUDENT表格的读权限，测试是否能查看STUDENT表格；

deny select on Student to STU\_wang

命令已成功完成。

select \* from Student

消息 229，级别 14，状态 5，第 1 行

拒绝了对对象 'Student' (数据库 'StudentDB'，架构 'dbo')的 SELECT 权限。

不能查看Student表格

1. 删除登录名“STU\_wang”；

drop user STU\_wang

命令已成功完成。

备份与恢复：

* 通过SQL语句实现以下功能

1. 设置简单恢复模式。

alter database StudentDB set recovery simple

命令已成功完成。

1. 对数据库做一次完整备份+1次差异备份。将数据库恢复到最后一次备份的时间。

use master

go

alter database StudentDB set recovery full with no\_wait

go

backup database StudentDB

filegroup = 'primary'

to disk = 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\primary.bak'with format

go

已为数据库 'StudentDB'，文件 'StudentDB' (位于文件 1 上)处理了 280 页。

已为数据库 'StudentDB'，文件 'StudentDB\_log' (位于文件 1 上)处理了 2 页。

BACKUP DATABASE...FILE=<name> 成功处理了 282 页，花费 0.082 秒(26.778 MB/秒)。

backup database StudentDB

filegroup = 'primary'

to disk = 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\primary.bak'with differential

go

已为数据库 'StudentDB'，文件 'StudentDB' (位于文件 2 上)处理了 40 页。

已为数据库 'StudentDB'，文件 'StudentDB\_log' (位于文件 2 上)处理了 2 页。

BACKUP DATABASE...FILE=<name> WITH DIFFERENTIAL 成功处理了 42 页，花费 0.027 秒(11.953 MB/秒)。

drop database StudentDB

命令已成功完成。

use master

go

restore database StudentDB

from disk = 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\primary.bak' with file =1,

nounload,norecovery,replace,stats = 5

go

已处理百分之 5。

已处理百分之 100。

已为数据库 'StudentDB'，文件 'StudentDB' (位于文件 1 上)处理了 280 页。

已为数据库 'StudentDB'，文件 'StudentDB\_log' (位于文件 1 上)处理了 2 页。

RESTORE DATABASE ... FILE=<name> 成功处理了 282 页，花费 0.080 秒(27.447 MB/秒)。

1. 设置完整恢复模式。

alter database StudentDB set recovery full

命令已成功完成。

1. 对数据库做一次完整备份+多次修改数据库中的内容+进行一次日志备份。删除数据库后，将数据库恢复。

use master

go

alter database StudentDB set recovery full with no\_wait

go

backup database StudentDB

filegroup = 'primary'

to disk = 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\primary.bak'with format

go

已为数据库 'StudentDB'，文件 'StudentDB' (位于文件 1 上)处理了 280 页。

已为数据库 'StudentDB'，文件 'StudentDB\_log' (位于文件 1 上)处理了 2 页。

BACKUP DATABASE...FILE=<name> 成功处理了 282 页，花费 0.135 秒(16.265 MB/秒)。

insert into Student

values ('S9','张三',19,'男','2004-09-09','人工智能')

(1 行受影响)

insert into SC

values ('S9','C1',97)

(1 行受影响)

use master

go

backup log StudentDB

to disk = 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\primary.bak'

go

已为数据库 'StudentDB'，文件 'StudentDB\_log' (位于文件 2 上)处理了 9 页。

BACKUP LOG 成功处理了 9 页，花费 0.020 秒(3.515 MB/秒)。

drop database StudentDB

命令已成功完成。

use master

go

restore database StudentDB

from disk = 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\primary.bak' with file =1,

nounload,norecovery,replace,stats = 5

go

已处理百分之 5。

已处理百分之 100。

已为数据库 'StudentDB'，文件 'StudentDB' (位于文件 1 上)处理了 280 页。

已为数据库 'StudentDB'，文件 'StudentDB\_log' (位于文件 1 上)处理了 2 页。

RESTORE DATABASE ... FILE=<name> 成功处理了 282 页，花费 0.104 秒(21.113 MB/秒)。

restore log StudentDB

from disk = 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\primary.bak' with file =1,

nounload,norecovery,replace,stats = 5

go

已为数据库 'StudentDB'，文件 'StudentDB' (位于文件 1 上)处理了 0 页。

已为数据库 'StudentDB'，文件 'StudentDB\_log' (位于文件 1 上)处理了 2 页。

已处理百分之 100。

RESTORE LOG 成功处理了 2 页，花费 0.017 秒(0.488 MB/秒)。

数据库实验六

1. 创建存储过程sp\_queryScore。将指定学生（姓名）是否选修指定课程(课程名)，如果有选修则输出“XX的XX课程考试成绩为X分。”如果成绩为空，则输出“XX的XX课程未考试”。如果不存在选课，则输出“XX未选修XX课程”。

create procedure sp\_queryScore

@sname char(10),@cname nchar(20),@score int output

as

set @score=(

select score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where sname = @sname and cname = @cname

)

if @sname not in

(

select sname

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where cname = @cname

)

print @sname + '未选修' + @cname+'课程'

if exists

(

select score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where sname = @sname and cname = @cname

)

print @sname+'的'+@cname+'课程考试成绩为'+cast(@score as char)

if exists

(

select score

from Student join SC on Student.sno = SC.sno

join Course on SC.cno = Course.cno

where sname = @sname and cname = @cname and score is null

)

print @sname +'的'+@cname +'课程未考核'

命令已成功完成。

1. 执行存储过程sp\_queryScore。

declare @score int

exec sp\_queryScore '张小明','高等数学',@score output

张小明 的高等数学 课程考试成绩为87

declare @score int

exec sp\_queryScore '张小明','程序设计',@score output

张小明 未选修程序设计 课程

1. 创建标量值函数f\_countStu。查询指定系别的总人数并返回。

create function f\_countStu(@dept nchar(10))

returns int

as

begin

declare @sum int

select @sum = count(distinct sno) from Student

where dept = @dept

return @sum

end

命令已成功完成。

select dept,dbo.f\_countStu(dept) as 总人数

from Student

1. 创建内联表值函数f\_sexStu。查询指定系别的系别名、男生人数和女生人数。

create function f\_sexStu(@dept nchar(10))

returns table

as

return (

select @dept as 系别 ,count(sno) as 男生人数 ,count(\*)-count(sno) as 女生人数

from Student

where dept = @dept and sex = '男'

)

命令已成功完成。

select \* from f\_sexStu('计算机')

go

create function f\_deptsexStu()

returns table

as

return (

select dept as 系别 ,count(sno) as 男生人数 ,count(\*)-count(sno) as 女生人数

from Student

where sex = '男'

group by dept

)

命令已成功完成。

select \* from f\_deptsexStu()

1. 在已有数据库基础上，创建后触发型触发器。约束每个学生在选某课之前，必须选修其先行课。并测试。

create trigger tri\_InsertSC on SC after Insert

as

begin

declare @sno varchar (20),@pno varchar(20)

select @sno = sno,@pno = pno from Course C join Inserted I on C.cno = I.cno

if @pno is not null and not exists(select \* from SC where sno = @sno and cno = @pno)

begin

print '未选修先行课'+@pno

rollback

end

else

print '顺利插入'

end

命令已成功完成。

测试：

insert into SC

values('C3','S1','90')

消息 3609，级别 16，状态 1，第 1 行

事务在触发器中结束。批处理已终止。

insert into SC

values('C2','S1','90')

顺利插入

（1行受影响）

1. 在已有数据库基础上，为Course表增加列“stuLimit（限选人数）”。创建前触发型触发器，约束每门课的选课人数不得超过stuLimit。并测试。

alter table Course

add stuLimit tinyint default(3)

命令已成功完成。

create trigger tri\_numLimit on SC instead of Insert

as

begin

declare @stuNum int,@stuLimit int

declare @cno varchar (20)

select @cno = C.cno,@stuLimit = stuLimit from Inserted I join Course C on C.cno = I.cno

select @stuNum=count(\*) from SC where cno=@cno

if @stuNum=@stuLimit

print'该课程已经满员'

else

begin

insert into SC select \*from inserted

print'选课成功！'

end

end

命令已成功完成。

测试：

创建一个视图进行存储每个课程的选课人数和限制人数

go

create view courseNumLimit

as

select max(sc.cno) as 课程号,count(\*)as 选课人数,max(stuLimit)as 限制人数

from SC join Course on sc.cno=Course.cno group by SC.cno

命令已成功完成。

go

select \* from courseNumLimit

insert into SC values('C2','S3','88')

insert into SC values('C1','S4','88')

该课程已经满员

(1行受影响)

1. 在第4题创建的f\_sexStu基础上。创建游标，按照“计算机系有X名男生和X名女生”的格式输出所有系别的学生信息。

declare @sdept varchar(20),@boyNum int, @girlNum int

declare cur cursor for select \*from f\_deptsexStu()

open cur

fetch next from cur into @sdept,@boyNum,@girlNum

while @@FETCH\_STATUS = 0

begin

print rtrim(@sdept)+'系有'+cast(@boyNum as varchar)+'名男生，'+cast(@girlNum as varchar)+'名女生。'

fetch next from cur into @sdept,@boyNum,@girlNum

end

close cur

deallocate cur

计算机系有1名男生，0名女生。

人工智能系有1名男生，0名女生。

软件工程系有2名男生，0名女生。

物联网系有1名男生，0名女生。