

实验报告 第四次实验

Hollow Man

一、实验十

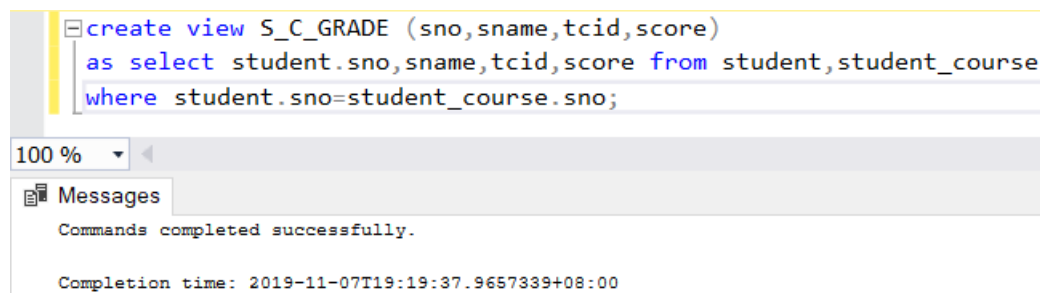
1. 问题 1

1)

使用如下 SQL 命令：

```
create view S_C_GRADE (sno,sname,tcid,score)
as select student.sno,sname,tcid,score from student,student_course
where student.sno=student_course.sno;
```

运行结果如下图所示：

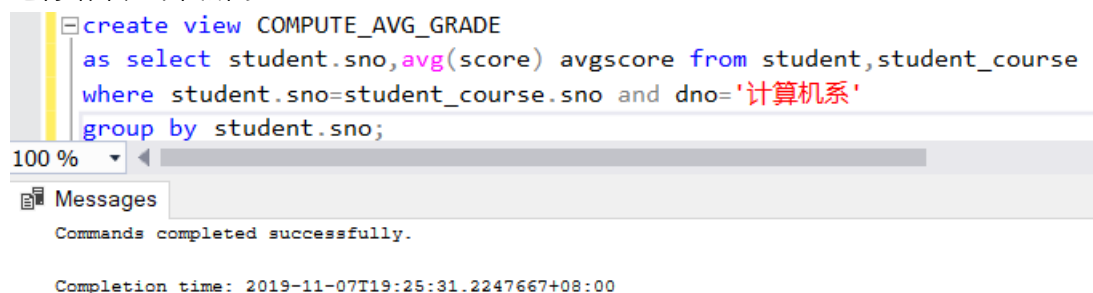


2)

使用如下 SQL 命令：

```
create view COMPUTE_AVG_GRADE
as select student.sno,avg(score) avgscore from student,student_course
where student.sno=student_course.sno and dno='计算机系'
group by student.sno;
```

运行结果如下图所示：

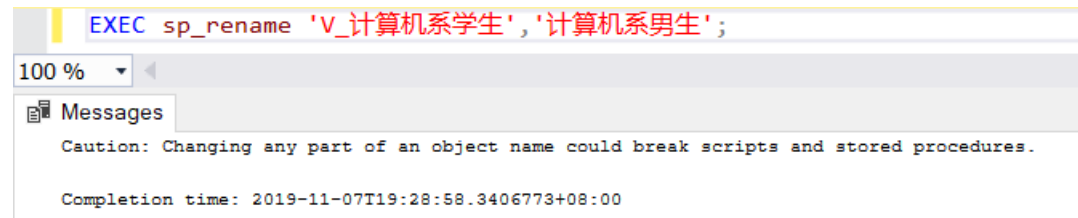


2. 问题 2

使用如下 SQL 命令：

```
EXEC sp_rename 'V_计算机系学生','计算机系男生';
```

运行结果如下图所示：



```
EXEC sp_rename 'V_计算机系学生', '计算机系男生';
```

100 %

Messages

Caution: Changing any part of an object name could break scripts and stored procedures.

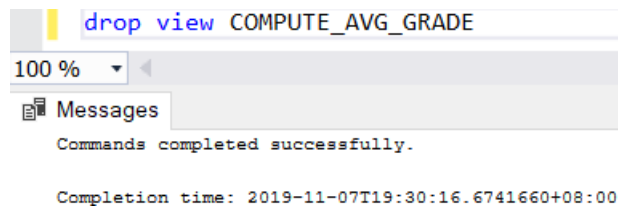
Completion time: 2019-11-07T19:28:58.3406773+08:00

3. 问题 3

使用如下 SQL 命令:

```
drop view COMPUTE_AVG_GRADE
```

运行结果如下图所示:



```
drop view COMPUTE_AVG_GRADE
```

100 %

Messages

Commands completed successfully.

Completion time: 2019-11-07T19:30:16.6741660+08:00

二、实验十一

1. 问题 1

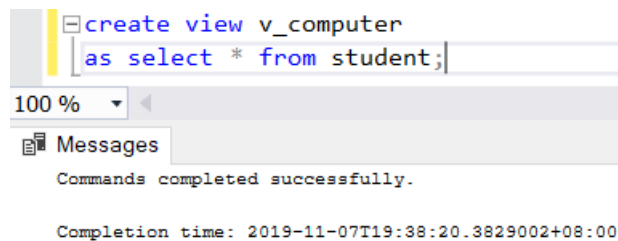
1)

使用如下 SQL 命令:

```
create view v_computer
```

```
as select * from student;
```

运行结果如下图所示:



```
create view v_computer  
as select * from student;
```

100 %

Messages

Commands completed successfully.

Completion time: 2019-11-07T19:38:20.3829002+08:00

2)

使用如下 SQL 命令:

```
create view V_S_C_G
```

as select student.sno 学号, sname 姓名, course.cno 课程号, cname 课程名, score 成绩

```
from student, student_course, course
```

```
where student.sno=student_course.sno and student_course.tcid=course.cno;
```

运行结果如下图所示:

```
create view V_S_C_G
as select student.sno 学号,sname 姓名,course.cno 课程号,cname 课程名,score 成绩
from student,student_course,course
where student.sno=student_course.sno and student_course.tcid=course.cno;
```

100 %

Messages

Commands completed successfully.

Completion time: 2019-11-07T20:24:34.6446499+08:00

3)

使用如下 SQL 命令:

因为给的表中没有学生年龄这一项,且 avg 函数不支持 smalldate 类型,所以将求年龄平均值改为求生日最小值:

```
create view V_NUM_AVG
as select count(dno) 人数,min(birthday) 平均年龄 from student
group by dno;
```

运行结果如下图所示:

```
create view V_NUM_AVG
as select count(dno) 人数,min(birthday) 平均年龄 from student
group by dno;
```

100 %

Messages

Commands completed successfully.

Completion time: 2019-11-07T19:50:38.8603682+08:00

4)

使用如下 SQL 命令:

```
create view V_YEAR(sname,birthyear)
as select sname, birthday from student;
```

运行结果如下图所示:

```
create view V_YEAR(sname,birthyear)
as select sname, birthday from student;
```

100 %

Messages

Commands completed successfully.

Completion time: 2019-11-07T19:55:22.4203377+08:00

5)

使用如下 SQL 命令:

```
create view V_AVG_S_G
as select student.sno 学号,sname 姓名,count(tcid) 选课门数,avg(score) 平均成绩
from student,student_course
where student.sno=student_course.sno
group by student.sno,sname
```

运行结果如下图所示:

```

create view V_AVG_S_G
as select student.sno 学号,sname 姓名,count(tcid) 选课门数,avg(score) 平均成绩
from student,student_course
where student.sno=student_course.sno
group by student.sno,sname

```

100 %

Messages

Commands completed successfully.

Completion time: 2019-11-07T20:03:56.7634714+08:00

6)

使用如下 SQL 命令:

```

create view V_AVG_C_G(课程号,选课人数,平均分)
as select tcid 课程号,count(sno) 选课人数,avg(score) 平均分
from student_course
group by tcid

```

运行结果如下图所示:

```

create view V_AVG_C_G(课程号,选课人数,平均分)
as select tcid 课程号,count(sno) 选课人数,avg(score) 平均分
from student_course
group by tcid

```

100 %

Messages

Commands completed successfully.

Completion time: 2019-11-07T20:07:31.4010911+08:00

2. 问题 2

2)

使用如下 SQL 命令:

```

select distinct sno,sname,score
from S_C_GRADE where score>90;

```

运行结果如下图所示:

```

select distinct sno,sname,score
from S_C_GRADE where score>90;

```

100 %

Results Messages

sno	sname	score
-----	-------	-------

3)

使用如下 SQL 命令:

```

select V_S_C_G.学号,V_AVG_S_G.姓名,V_S_C_G.课程名,V_S_C_G.成绩

```

运行结果如下图所示:

4)

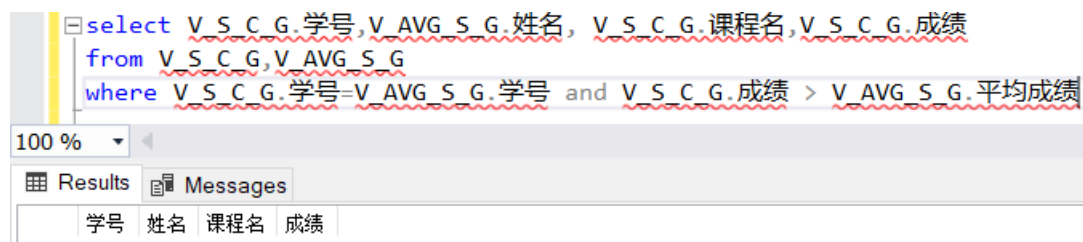
使用如下 SQL 命令:

```

select V_S_C_G.学号,V_AVG_S_G.姓名,V_S_C_G.课程名,V_S_C_G.成绩
from V_S_C_G,V_AVG_S_G
where V_S_C_G.学号=V_AVG_S_G.学号 and V_S_C_G.成绩 > V_AVG_S_G.平均成绩

```

运行结果如下图所示：



3. 问题 3

注：因定义的视图中无 V_IS, 所以本问中同一使用正确的视图 V_Computer

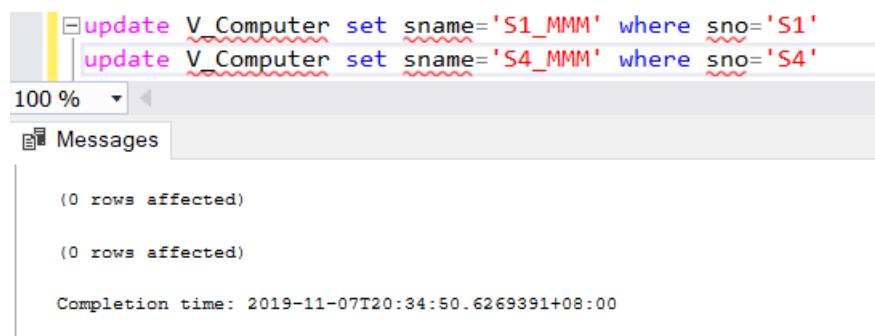
1)

使用如下 SQL 命令：

```
update V_Computer set sname='S1_MMM' where sno='S1'
```

```
update V_Computer set sname='S4_MMM' where sno='S4'
```

运行结果如下图所示：



2)

使用如下 SQL 命令：

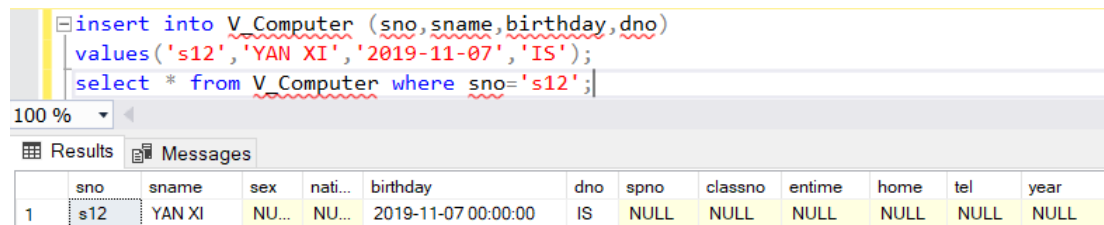
注：因无年龄对应的列，因而将年龄 19 改为生日日期 2019-11-07：

```
insert into V_Computer (sno,sname,birthday,dno)
```

```
values('s12','YAN XI','2019-11-07','IS');
```

```
select * from V_Computer where sno='s12';
```

运行结果如下图所示：



3)

不能实现。因为 V_AVG_S_G 视图的字段来自聚集函数，含有 GROUP BY 字句，此视图不允许更新。

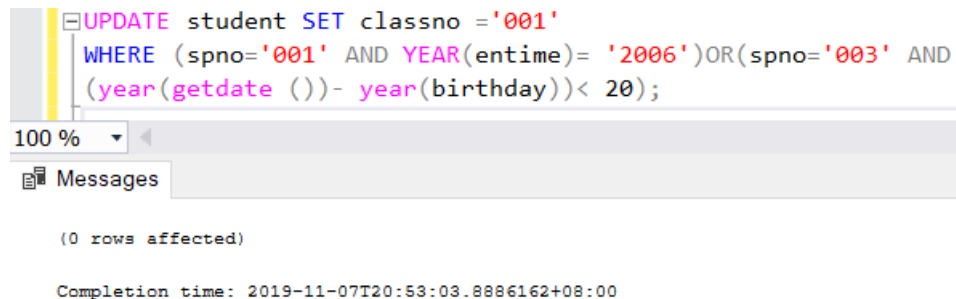
三、实验十二

1. 问题 1

使用如下 SQL 命令：

```
UPDATE student SET classno = '001'
WHERE (spno='001' AND YEAR(entime)= '2006')OR(spno='003' AND
(year(getdate ())- year(birthday))< 20);
```

运行结果如下图所示：

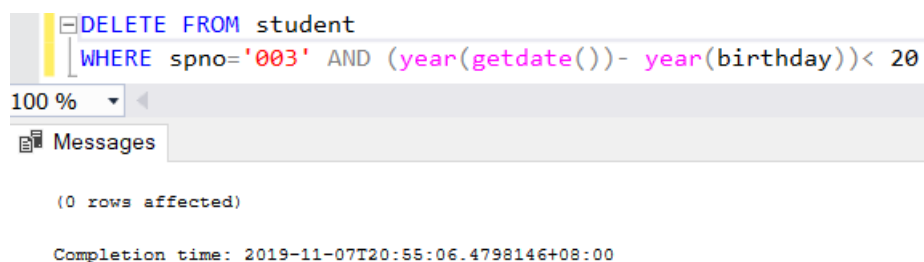


2. 问题 2

使用如下 SQL 命令：

```
DELETE FROM student
WHERE spno='003' AND (year(getdate())- year(birthday))< 20
```

运行结果如下图所示：

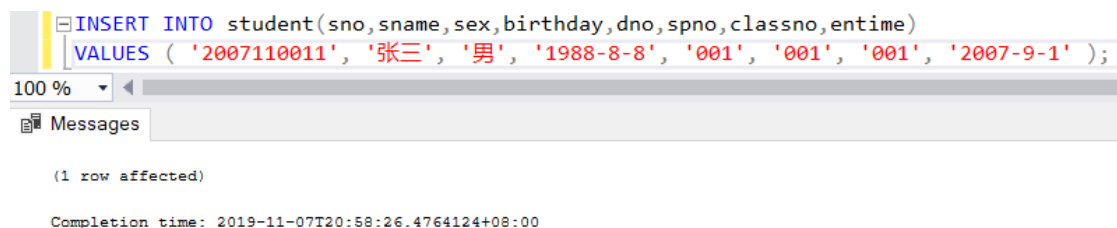


3. 问题 3

使用如下 SQL 命令：

```
INSERT INTO student(sno,sname,sex,birthday,dno,spno,classno,entime)
VALUES ( '2007110011', '张三', '男', '1988-8-8', '001', '001', '001', '2007-9-1' );
```

运行结果如下图所示：



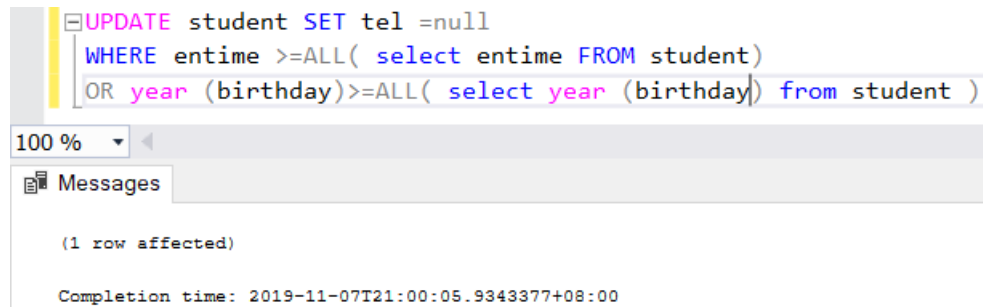
4. 问题 4

使用如下 SQL 命令：

```
UPDATE student SET tel =null
WHERE entime >=ALL( select entime FROM student)
```

```
OR year (birthday)>=ALL( select year (birthday) from student)
```

运行结果如下图所示:



```
UPDATE student SET tel =null
WHERE entime >=ALL( select entime FROM student)
OR year (birthday)>=ALL( select year (birthday) from student )
```

100 %

Messages

(1 row affected)

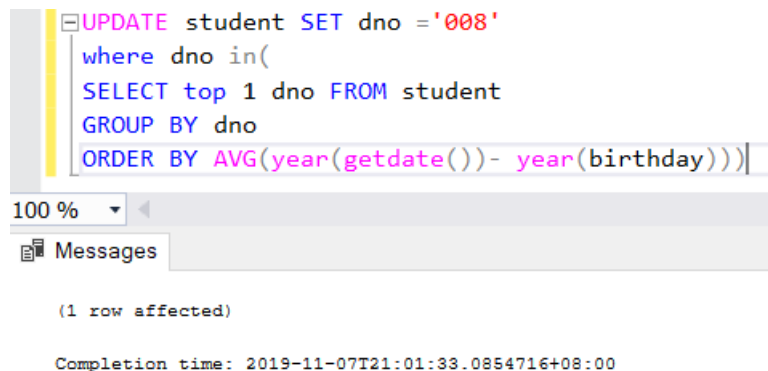
Completion time: 2019-11-07T21:00:05.9343377+08:00

5. 问题 5

使用如下 SQL 命令:

```
UPDATE student SET dno ='008'
where dno in(
SELECT top 1 dno FROM student
GROUP BY dno
ORDER BY AVG(year(getdate())- year(birthday)))
```

运行结果如下图所示:



```
UPDATE student SET dno ='008'
where dno in(
SELECT top 1 dno FROM student
GROUP BY dno
ORDER BY AVG(year(getdate())- year(birthday)))
```

100 %

Messages

(1 row affected)

Completion time: 2019-11-07T21:01:33.0854716+08:00

四、实验十三

1. 问题 1

代码如下:

```
DECLARE @year INT
SELECT @year =2014
if
@year %4 =0 and @year %100 <> 0 print CONVERT( CHAR( 4), @year )+ ' 是闰年 '
else print CONVERT( CHAR( 4 ), @year )+ ' 不是闰年 '
```

运行结果如下:

```

DECLARE @year INT
SELECT @year =2014
if
@year %4 =0 and @year %100 <> 0 print CONVERT( CHAR( 4), @year )+ ' 是闰年 '
else print CONVERT( CHAR( 4 ), @year )+ ' 不是闰年 '

```

100 %

Messages

2014 不是闰年

Completion time: 2019-11-07T21:02:37.0567672+08:00

2. 问题 2

1)

代码如下:

```

DECLARE @SUM INT,@I INT
SELECT @I=1,@SUM=0
WHILE @I<=100
BEGIN
    SELECT @SUM=@SUM+@I
    SELECT @I=@I+1
END
PRINT '1...100 的和为: '+CONVERT(CHAR(4),@SUM)

```

运行结果如下:

```

DECLARE @SUM INT,@I INT
SELECT @I=1,@SUM=0
WHILE @I<=100
BEGIN
    SELECT @SUM=@SUM+@I
    SELECT @I=@I+1
END
PRINT '1...100的和为: '+CONVERT(CHAR(4),@SUM)

```

100 %

Messages

1...100的和为: 5050

Completion time: 2019-11-07T21:03:38.9609759+08:00

可知, While 循环语句块为标志 BEGIN ... END 之间的部分

2)

代码如下:

```

DECLARE @SUM INT , @I INT
SELECT @I=1 , @SUM=1
WHILE @I<= 10
BEGIN
    SELECT @SUM=@SUM* @I
    SELECT @I=@I+1
END
PRINT '10 的阶乘的结果为: ' +CONVERT(CHAR(10), @SUM)

```


运行结果如下：

```
DECLARE @SUM INT , @I INT
SELECT @I=1 , @SUM=1
WHILE @I<= 10
BEGIN
    SELECT @SUM=@SUM* @I
    SELECT @I=@I+1
END
PRINT '10的阶乘的结果为: ' +CONVERT(CHAR(10), @SUM)
```

100 %

Messages

10的阶乘的结果为: 3628800

Completion time: 2019-11-07T21:06:08.4279995+08:00

3)

代码如下：

```
SELECT
main . number
FROM
master .. spt_values AS main
WHERE
type = 'P'
AND number BETWEEN 3 AND 100
AND NOT EXISTS (
SELECT
1
FROM
master .. spt_values AS sub
WHERE
type = 'P'
AND sub.number BETWEEN 2 AND main . number - 1
AND main.number % sub.number = 0 )
```

运行结果如下：

```
SELECT
main . number
FROM
master .. spt_values AS main
WHERE
type = 'P'
AND number BETWEEN 3 AND 100
AND NOT EXISTS (
SELECT
1
FROM
master .. spt_values AS sub
WHERE
type = 'P'
AND sub.number BETWEEN 2 AND main . number - 1
AND main.number % sub.number = 0 )
```

100 %

Results Messages

	number
1	3
2	5
3	7
4	11
5	13
6	17
7	19
8	23
9	29
10	31
11	37
12	41
13	43
14	47
15	53
16	59
17	61
18	67
19	71

3. 问题 3

1)

代码如下:

```
select 书名, 出版社, 作者,
case
    when 定价>50 then '定价太高, 不适合作教材'
    else '定价' + CAST(定价 as varchar(5)) + ', 可以作教材'
end 可否作为教材
from 图书
```

运行结果如下:

```

select 书名, 出版社, 作者,
       case
         when 定价>50 then '定价太高, 不适合作教材'
         else '定价' + CAST(定价 as varchar(5)) + ', 可以作教材'
       end 可否作为教材
from 图书

```

	书名	出版社	作者	可否作为教材
1	数据结构	机械工业出版社	王民	定价太高, 不适合作教...
2	计算机应用	机械工业出版社	张建平	定价20.00, 可以作教材
3	数据库技术	电子工业出版社	王敏	定价15.00, 可以作教材
4	C语言	电子工业出版社	谭浩强	定价25.00, 可以作教材
5	应用文写作	中国人民大学出版...	张锦芯	定价25.00, 可以作教材
6	管理学	高等教育出版社	Robison	定价15.00, 可以作教材
7	工业管理	机械工业出版社	Fayol	定价太高, 不适合作教...
8	线性代数	机械工业出版社	李平	定价50.00, 可以作教材
9	公司的概念	机械工业出版社	Durark	定价14.00, 可以作教材
10	统计学	机械工业出版社	徐新国	定价15.00, 可以作教材

2)

代码如下:

```

SELECT student.sno, sname, cname,
CASE
WHEN student_course . score >= 90 THEN ' 优'
when student_course . score >= 80 and student_course . score <= 90 THEN '良'
,
when student_course . score >= 70 and student_course . score <= 80 THEN '中'
,
when student_course . score >= 60 and student_course . score <= 70 THEN '及格'
else '不及格'
END GRADE
FROM student_course, course, student
WHERE student_course.tcid=course.cno AND
student_course.sno =student.sno

```

运行结果如下:

```

SELECT student.sno, sname, cname,
CASE
WHEN student_course . score >= 90 THEN ' 优'
when student_course . score >= 80 and student_course . score <= 90 THEN '良'
when student_course . score >= 70 and student_course . score <= 80 THEN '中'
when student_course . score >= 60 and student_course . score <= 70 THEN '及格'
else '不及格'
END GRADE
FROM student_course, course, student
WHERE student_course.tcid=course.cno AND
student_course.sno =student.sno

```

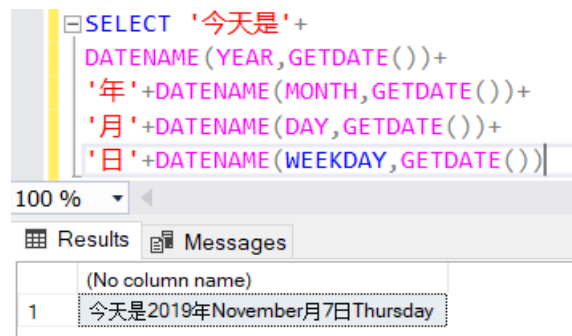
sno	sname	cname	GRADE
1	王民	数据结构	不及格
2	张建平	计算机应用	良
3	王敏	数据库技术	良
4	谭浩强	C语言	良
5	张锦芯	应用文写作	良
6	Robison	管理学	良
7	Fayol	工业管理	不及格
8	李平	线性代数	不及格
9	Durark	公司的概念	及格
10	徐新国	统计学	及格

4. 问题 4

代码如下：

```
SELECT '今天是'+  
DATENAME(YEAR,GETDATE())+  
'年'+DATENAME(MONTH,GETDATE())+  
'月'+DATENAME(DAY,GETDATE())+  
'日'+DATENAME(WEEKDAY,GETDATE())
```

运行结果如下：



通过系统函数，可以查询一些关于系统的重要信息。