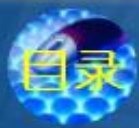




T-SQL程序设计

学习目标

- 掌握如何定义变量并赋值。
- 掌握如何输出显示数据。
- 掌握逻辑控制语句。
- 理解**SQL**中批处理的概念。





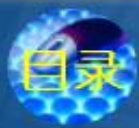
T-SQL程序设计

知识重点

- 如何定义变量并赋值。
- 逻辑控制语句。

知识难点

- 逻辑控制语句。





5.1 知识准备

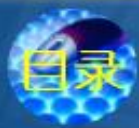
5.1.1 SQL标识符

在T-SQL语句中，不使用分隔标识符分开的标识符为常规标识符。常规标识符的命名规则如下：

首字符必须以字母、下划线（_）、@、#等符号开头。

标识符不能是**T-SQL**的保留字。

不允许嵌入空格或其他特殊字符。

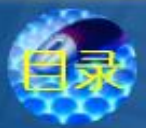




5.1.1 SQL标识符

分隔符标识符

代字号~	连字符-
惊叹号!	左括号{
百分号%	右括号}
插入号^	撇号'
“与”符号&	句点.
左圆括号(反斜杠\
右圆括号)	重音符号`

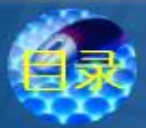




5.1.2 常量与变量

常量

常量是表示特定数据值的符号。常量的格式取决于它所表示的值的数据类型。常量根据不同的数据类型分为字符串常量、二进制常量、bit常量、datetime常量、integer常量、decimal常量、float和real常量、money常量、uniqueidentifier常量。





5.1.2 常量和变量

局部变量

局部变量的名称必须以标记@作为前缀。

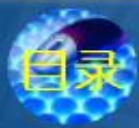
声明局部变量的语句为：

```
DECLARE variable_name DataType
```

其中，variable_name为局部变量的名称，DataType为数据类型。

全局变量

T-SQL中的所有全局变量都使用两个@标志作为前缀。





5.1.2 常量和变量

局部变量的赋值，set与select
语法：

Set @变量名=值

select @变量名=值

●例1： Set的使用

●declare @i int

●set @i=2

●print @i

例2： select 的使用

●declare @x varchar(10)

●select @x='123456'

●print @x

例3： select 的使用

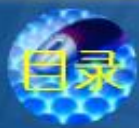
●declare @name varchar(6)

●select

@name=student_tab.stud_name from
student_tab where stud_id='101'

●Print @name

●Select @name

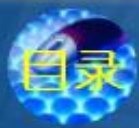




5.1.2 常量和变量

全局变量

变量	含义
<code>@@ERROR</code>	最后一个T-SQL 错误的错误号
<code>@@IDENTITY</code>	最后一次插入的标识值
<code>@@LANGUAGE</code>	当前使用的语言的名称
<code>@@MAX_CONNECTIONS</code>	可以创建的同时连接的最大数目
<code>@@ROWCOUNT</code>	受上一个SQL 语句影响的行数
<code>@@SERVERNAME</code>	本地服务器的名称
<code>@@SERVICENAME</code>	该计算机上SQL 服务器的名称
<code>@@TIMETICKS</code>	当前计算机上每刻度的微秒数
<code>@@TRANSCOUNT</code>	当前连接打开的事务数
<code>@@VERSION</code>	SQL Server 的版本信息





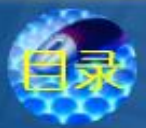
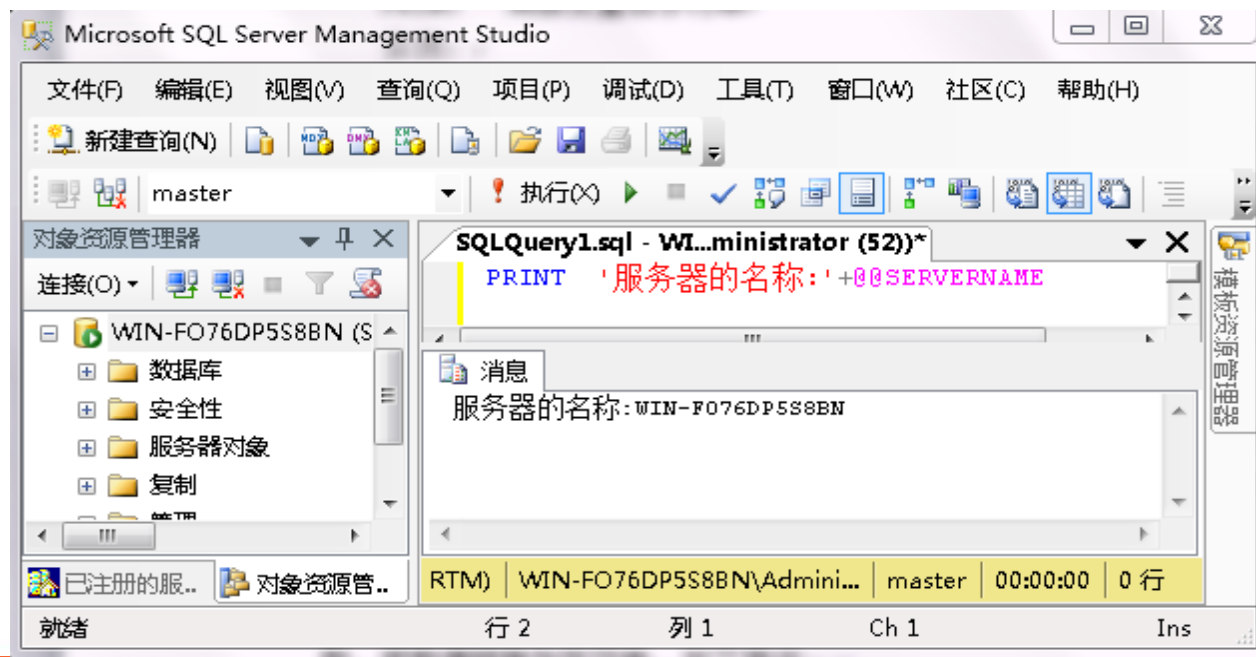
5.2 项目实施

5.2.1 任务1：输出语句的使用

子任务1：PRINT语句的使用

显示数据库服务器的名称。

PRINT '服务器的名称:'+'@@SERVERNAME'。

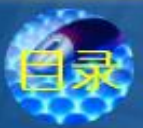
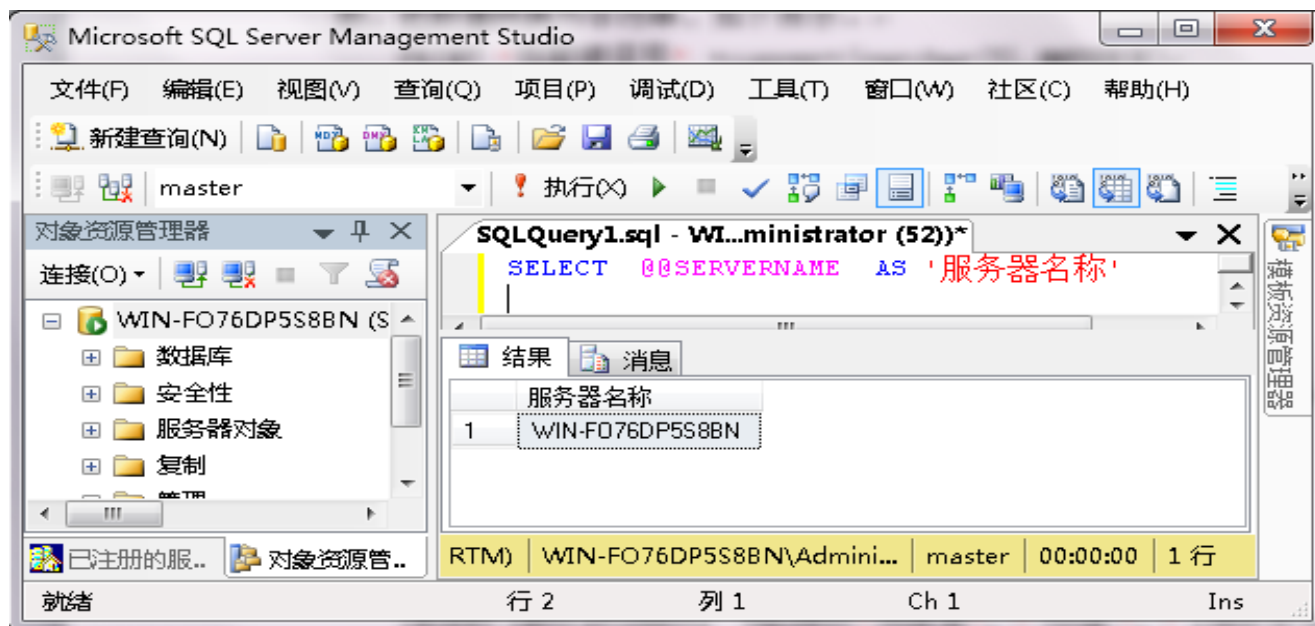


5.2.1 任务1：输出语句的使用

子任务2：查询语句的特殊应用

SELECT 局部变量AS自定义列名

该方法就是查询语句的特殊应用,用于输出显示处理的数据结果。例如：
SELECT @@SERVERNAME AS '服务器名称'



5.2.2 任务2：逻辑控制语句的使用

子任务1：IF…ELSE条件语句的使用

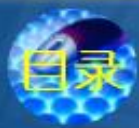
IF-ELSE语句的语法为：

IF（条件）

语句或语句块

ELSE

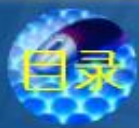
语句或语句块



5.2.2 任务2：逻辑控制语句的使用

例1：查询课程表中是否开设计算机专业英语的课程，如果有，则查询该课程的详细信息（用T-SQL实现）

```
if exists (select * from course_tab where course_name='计算机专业英语')
    select * from course_tab
    where course_name='计算机专业英语'
else
    print'没有开设计算机专业英语这门课程'
```

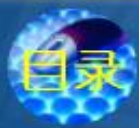




5.2.2 任务2：逻辑控制语句的使用

例2：查询课程计算机基础考试的平均分，如果平均分在75以上则显示为“成绩优秀”，并显示前3名学生的考试信息，如果平均分在75分以下，则显示“成绩一般”，并显示成绩最差的学生学号，课程号和成绩

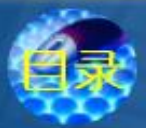
```
declare @avgscore float
declare @cid varchar(6)
select @cid=course_tab.course_id from course_tab where course_name='计算机基础'
select @avgscore=avg(score)
from score_tab where course_id=@cid
print '计算机基础的平均分为'+convert (varchar(6),@avgscore)
if(@avgscore>75)
begin
    print'本班计算机基础课程考试成绩为优秀，前三名的成绩为‘
select top 3*
    from score_tab
    where course_id=@cid
    order by score desc
end
```



5.2.2 任务2：逻辑控制语句的使用

例2：查询课程计算机基础考试的平均分，如果平均分在75以上则显示为“成绩优秀”，并显示前3名学生的考试信息，如果平均分在75分以下，则显示“成绩一般”，并显示成绩最差的学生学号，课程号和成绩
续：

```
else
begin
print'本班计算机基础课程考试成绩为一般，最差的学生成绩是'
select top 1 *
from score_tab
where course_id=@cid
order by score
end
```



5.2.2 任务2：逻辑控制语句的使用

子任务2：WHILE 循环语句的使用

WHILE循环语句的语法：
WHILE（条件）
 语句或语句块
[BREAK]

5.2.2 任务2：逻辑控制语句的使用

--while的用法

--例：求 $1+2+3+\dots+100$ 的值

```
declare @n int,@sum int
```

```
set @n=0
```

```
set @sum=0
```

```
while @n<=100
```

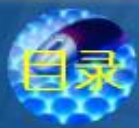
```
begin
```

```
    set @sum=@sum+@n
```

```
    set @n=@n+1
```

```
end
```

```
print'1+2+3+\dots+100=' +cast(@sum as varchar(4))
```





子任务3：CASE多分支语句的使用

CASE语句的语法如下：

CASE

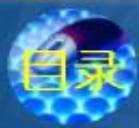
WHEN 条件1 THEN 结果1

WHEN 条件2 THEN 结果2

.....

[ELSE 其他结果]

END



5.2.2 任务2：逻辑控制语句的使用

--case的用法

--例1：采用五级制显示学生的成绩（包含的列名有姓名，课程名和成绩评定）

```
select student_tab.stud_name as 姓名 ,course_tab.course_name as 课程名,score_tab.score as 成绩,
```

```
case
```

```
    when score>=90          then '优秀'
```

```
    when score between 80 and 89 then '良好'
```

```
    when score between 70 and 79 then '中等'
```

```
    when score between 60 and 69 then '及格'
```

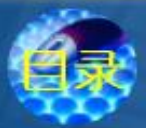
```
    else '不及格'
```

```
end as 成绩等级
```

```
from student_tab,course_tab,score_tab
```

```
where student_tab.stud_id=score_tab.stud_id and
```

```
    course_tab.course_id=score_tab.course_id
```





5.2.3 任务3：批处理语句的使用

Use Master

GO

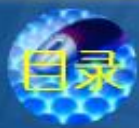
GO关键字标志着批处理的结束。可以把多条语句组成一个执行计划，然后再执行。例如：

```
SELECT * FROM students
```

```
SELECT * FROM score
```

```
SELECT * FROM course
```

GO





5.4 项目总结

和C语言非常相似，变量的使用也是先声明，然后再赋值。局部变量前必须有“@”前缀，全局变量必须有两个“@”前缀。

变量的赋值有两种方式：使用**SET**语句或**SELECT**语句。

输出结果也有两种方式：**PRINT** 语句和**SELECT**语句。

控制流语句提供了条件操作所需的顺序和逻辑。

语句块使用**BEGIN...END**。

批处理可以提高语句执行的效率，批处理结束的标志是“**GO**”。

