P1

记录型：学生(学号 char(8),姓名 char(6),性别 char(2),年龄 tinyint,系别varchar(20))。

记录值：('210201','李欣','女',18,'财务会计系')。

P23

（4）像集[[1]](#footnote-1)（Images Set）。在关系R(X，Z)中，X和Z为属性组，给出定义：当t[X]=x时，（与之连接的）属性组Z的诸分量的集合称为（属性组X取值x时的）像集Zx。记作：

Zx={t[Z]|t∈R∧t[X]=x}

#### 注意：像集Zx可以理解为分量值x在属性组Z上的诸多映像的集合，即像集是x的不同摆拍映像的集合。

属性组X各分量的像集如下。

当t[X]= x1时，x1在属性组Z上的像集：Zx1={z1,z2,z3}；

当t[X]= x2时，x2在属性组Z上的像集：Zx2={z1,z2}；

当t[X]= x3时，x3在属性组Z上的像集：Zx3={z1}。

P31页第（7）连接

R∞S={t(5)|(∃u)(∃v)(R(u)∧S(v)∧t[1]=u[1]∧t[2]=u[2]∧t[3]=u[3]∧t[4]=v[1]∧t[5]=v[2]∧u[3]=v[3])}

P32

#### 注意：求解元组演算时，一定要分清已知关系的元组变量和所求关系的元组变量，并根据两者之间各个分量的θ条件，实现所求关系的元组（集合）。

P33页第2行

满足条件y>20和z>1600的元组构成。运算结果如表2-29所示。

#### 注意：求解域演算时，一定要分清已知关系的域变量和所求关系的域变量，并根据两者之间各个分量的θ条件，实现所求关系的元组（集合）。

P50-P51

4．有关系模式P(C,S,T,R,G)，根据语义有如下函数依赖集：F{C→T,ST→R,TR→C,SC→G}。下列属性组中的哪（ ）个（些）是关系模式P的候选码。

15．设有关系模式W(工号,姓名,工种,定额)，将其规范化到第三范式正确的答案是（ ）。

A．W1(工号,姓名)，W2(工种,定额)

B．W1(工号,工种,定额)，W2(工号,姓名)

C．Wl(工号，姓名，工种)，W2(工种，定额)

D．以上都不对

P75页

#### 1．用户表的表结构

数据库JXGL中存在如下几个用户表，各表的逻辑结构分别定义如表5-10~5-15所示。

表5-10 班级信息表（班级） 表5-11 选修信息表（选修）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 |  | 字段名 | 数据类型 | 备注 |
| 班级号 | char(6) | 主键 |  | 成绩编码 | int | 标识规范 |
| 班级名称 | varchar(20) | 唯一约束 |  | 学号 | char(8) | 非空、外键 |
| 班级人数 | tinyint |  |  | 课程号 | char(2) | 非空、外键 |
| 学制 | char(1) | 默认值为4 |  | 成绩 | numeric(5,2) | 100≥成绩≥0 |
| 招生性质 | char(4) |  |  | 备注 | varchar(MAX) |  |

表5-12 学生信息表（学生） 表5-13 教师信息表（教师）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 |  | 字段名 | 数据类型 | 备注 |
| 学号 | char(8) | 非空，主键 |  | 工号 | char(6) | 非空，主键 |
| 姓名 | char(6) | 非空 |  | 姓名 | char(6) | 非空 |
| 性别 | char(2) | 默认值男 |  | 性别 | char(2) | 默认值男 |
| 出生日期 | datetime |  |  | 出生日期 | datetime |  |
| 总分 | int |  |  | 工作日期 | datetime |  |
| 籍贯 | char(4) | 默认值安徽 |  | 职称 | char(6) |  |
| 备注 | varchar(MAX) |  |  | 基本工资 | int |  |
| 班级号 | char(6) | 外键 |  | 婚否 | bit | 默认值0 |

表5-14 课程信息表（课程） 表5-15 授课信息表（授课）

| 字段名 | 数据类型 | 备注 |  | 字段名 | 数据类型 | 备注 |
| --- | --- | --- | --- | --- | --- | --- |
| 课程号 | char(2) | 主键 |  | 工号 | char(6) | 非空，外键 |
| 课程名称 | varchar(20) | 唯一约束 |  | 课程号 | char(2) | 非空，外键 |
| 课程类型 | char(4) |  |  | 班级号 | char(6) | 非空，外键 |
| 学时 | smallint | 学时≥0 |  | 课酬 | int | 12000≥课酬≥0 |
| 学分 | tinyint | 学分≥0 |  | 学期 | char(1) |  |
| 备注 | varchar(MAX) |  |  | 评价 | varchar(MAX) |  |

（2）单击释放后，弹出“检查约束”对话框，在约束编辑列表框中选择“表达式”选项，然后在其右侧文本框中输入“课酬>=0 and 课酬<=12000”，如图5-21所示。

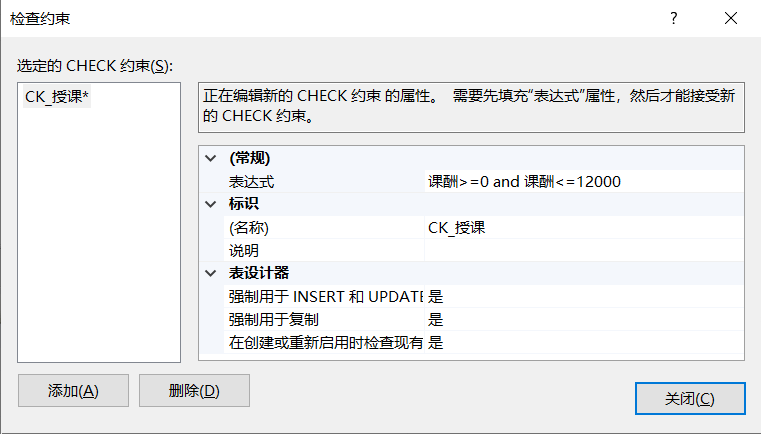


图5-21 “检查约束”对话框

全文 中

left(学号,6) 修改 为 班级号

P79

【例5-3】 使用T-SQL语句设计表“授课”，但不设置其相关约束。

use jxgl

go

create table 授课

(工号 char(6),

课程号 char(2),

班级号 char(6),

课酬 int,

学期 char(1),

评价 varchar(MAX))

【例5-4】 使用T-SQL语句设计表“教师”，仅设置列“工号”为主键，不设置其他约束。

use jxgl

go

create table 教师

(工号 char(6) constraint pk\_教师\_工号 primary key

,姓名 char(6)

,性别 char(2)

,出生日期 datetime

,工作日期 datetime

,职称 char(6)

,基本工资 int

,婚否 bit)

#### 注意：constraint pk\_教师\_工号可省略，省略时会由系统产生一个随机的约束标识名。

【例5-5】 使用T-SQL语句设计表“学生”，仅设置学号为非空、主键，姓名非空，性别默认值为“男”，但不设置籍贯的默认约束。

use jxgl

go

create table 学生(

学号 char(8) not null constraint pk\_学生 primary key,

姓名 char(6) not null,

性别 char(2) constraint df\_学生\_性别 default '男',

出生日期 datetime,

总分 int,

籍贯 char(4),

备注 varchar(MAX),

班级号 char(6))

在“对象资源管理器”窗格中可以查看“学生”表的相关信息，如图5-9所示。

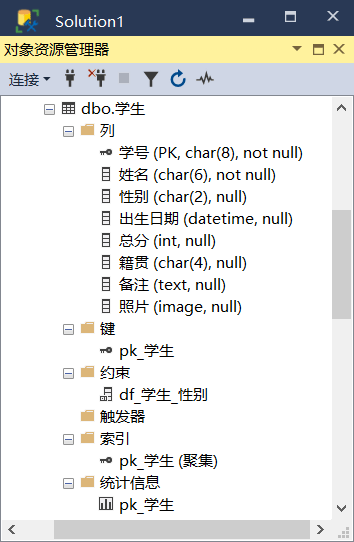
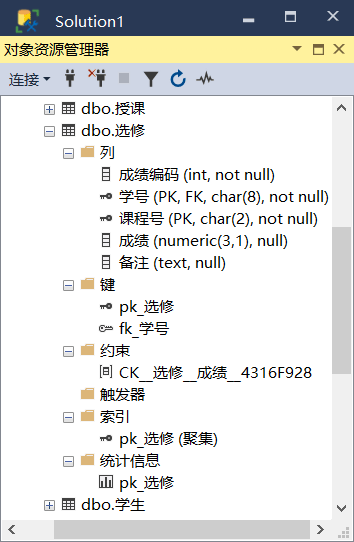
 

图5-9 “学生”表信息 图5-10 “选修”表信息

【例5-6】 使用T-SQL语句设计表“选修”，并设置成绩编码为标识规范列，学号为非空、外键，约束于“学生”表“学号”列，学号和课程号共同建立主键，成绩检查约束范围为0~100。

use jxgl

go

create table 选修(

成绩编码 int identity(1,1)

,学号 char(8) not null constraint fk\_学号 foreign key references 学生(学号)

,课程号 char(2) not null

,constraint pk\_选修 primary key(学号,课程号)

,成绩 numeric(3,1) check(成绩>=0 and 成绩<=100) /\*check约束也可以设置约束名\*/

,备注 varchar(MAX))

在“对象资源管理器”窗格中可以查看“选修”表的相关信息，如图5-10所示。

P103

【例6-28】 在“选修”表中查询选修了学号“19020102”同学选修课程的选修信息（比较【例6-34】）

select \* from 选修 where 课程号 in (select 课程号 from 选修 where 学号='19020102')

P104

【例6-34】 查询至少选修了学号为“19020102”同学全部选修课程的学生“姓名”（比较【例6-28】）。

分析：查询这样的学生x，不存在这样的课程y，是学号“19020102”同学选修的课程，而学生x没选修的课程。逻辑蕴涵表达式：(∀y)p→ q，其中p表示学号“19020102”的学生的全部选修课程y，q表示学生x的选修课程y。

select 姓名 from 学生 a /\*查询学生x\*/

where not exists (select \* from 选修 b where b.学号='19020102' /\*不存在课程y\*/

and not exists (select \* from 选修 c /\*x不选课程\*/

where c.学号=a.学号 and c.课程号=b.课程号))

P115页

#### 2．局部变量的声明

声明局部变量的语法格式如下：

格式：declare {@局部变量名 [as] 数据类型} [,…n]

功能：声明一个指定数据类型的局部变量。

说明：

①局部变量名必须符合标识符命名规则；

②数据类型可以是系统数据类型，也可以是用户自定义数据类型，但不能定义为text，ntext或image数据类型，如有需要，还需指定数据宽度及小数精度；

③声明多个局部变量名时，各变量名之间用逗号隔开；

④局部变量声明后，系统自动初始化赋值为null，局部变量声明时可以同时赋值。

P116

### 7.1.3 表达式的运算符

表达式是指用运算符将运算对象（操作数）连接起来的一个有意义的字符序列组合，而运算符则是用来连接和操作运算对象（操作数）的符号。T-SQL语句提供了7类运算符：算术运算符、比较运算符、按位逻辑运算符、逻辑运算符、字符串连接运算符、赋值运算符和一元运算符。

#### 1．算术运算符

算术运算符用于数值数据的数学运算，算术运算符及其含义如表7-2所示。

表7-2 算术运算符及其含义

|  |  |  |
| --- | --- | --- |
| 算术运算符 | 含义 | 数据类型 |
| +、-、\*、/ | 加、减、乘、除 | int、smallint、tinyint、decimal、float、real、money、smallmoney |
| % | 求余 | int、smallint、tinyint |

#### 2．比较运算符

比较运算符用于比较两个表达式的值是否相同，如果值相同则为true，否则为false，当参与比较的操作数含有null值时，结果为unknown。比较运算符及其含义如表7-3所示。

表7-3 比较运算符及其含义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 关系运算符 | 含义 | 关系运算符 | 含义 | 关系运算符 | 含义 |
| = | 等于 | >= | 大于或等于 | != | 不等于 |
| > | 大于 | <= | 小于或等于 | !< | 不小于 |
| < | 小于 | <> | 不等于 | !> | 不大于 |

注意：比较运算符不能用于text、ntext、image数据类型表达式的运算。另外，有时也把all、any、some、between…and、in、like等运算符当作逻辑运算符的一部分。

3．按位逻辑运算符

按位逻辑运算符用于整型数据和bit、binary或varbinary数据类型之间的按位逻辑运算，其返回值数据类型是较大数据类型，按位逻辑运算符及其含义如表7-4所示。

表7-4 按位逻辑运算符及其含义

|  |  |
| --- | --- |
| 位运算符 | 含义 |
| & | 按位与，双目运算，参与运算的两个位值都是1时结果为1，否则为0 |
| | | 按位或，双目运算，参与运算的两个位值都是0时结果为0，否则为1 |
| ^ | 按位异或，双目运算，参与运算的两个位值不同时结果为1，否则为0 |
| ~ | 按位取非，单目运算，即~1=0，~0=1 |

注意：

（1）在按位逻辑运算的双目运算中，只有一个表达式可以是binary或varbinary数据类型。

（2）在按位逻辑运算的双目运算中，如果左右表达式具有不同的整型数据，则将较小数据类型的参数转换为较大数据类型的参数；

（3）在按位逻辑运算的过程中，如果一个表达式是整型数据，则先将整型数据转换为二进制数据，然后再进行按位逻辑运算。

#### 4．逻辑运算符

逻辑运算符用于测试某些逻辑表达式真实性的运算，其返回值为true、false或unknown。逻辑运算符及其含义如表7-6所示。

表7-6 逻辑运算符及其含义

|  |  |
| --- | --- |
| 逻辑运算符 | 含义 |
| not | 非运算，单目运算，对逻辑表达式的值取反，即not(true)=false，not(false)=true |
| and | 与运算，双目运算，参与运算的两个逻辑表达式的值都是true时才为true，否则为false |
| or | 或运算，双目运算，参与运算的两个逻辑表达式的值都是false时才为false，否则为true |

#### 5．字符串连接运算符

字符串连接运算符用于将两个字符串连接成一个新字符串的运算，其返回值是一个字符串。字符串连接运算符只有一个，即加号。

#### 6．赋值运算符

赋值运算符用于将表达式的值赋给变量的运算。赋值运算符只有一个，即等号。

#### 7．一元运算符

一元运算符用于计算只有一个数值数据对象的运算，其返回值是数值数据。一元运算符及其含义如表7-7所示。

表7-7 一元运算符及其含义

|  |  |
| --- | --- |
| 一元运算符 | 含义 |
| + | 表示一个正数 |
| - | 表示一个负数 |
| ~ | 求一个数值数据的补码 |

### 7.1.4 运算符的优先级

当混合使用多种运算符构成一个复杂的表达式时，表达式中有括号则先算括号内，再算括号外；无括号时，运算符的优先级决定了运算的先后顺序，并影响计算的结果。运算符的优先级从高到低排列顺序如表7-8所示。

表7-8 运算符的优先级排序

|  |  |
| --- | --- |
| 运算符 | 优先级 |
| ~（按位非） | 1 |
| \*（乘）、/（除）、%（模） | 2 |
| +（正）、-（负）、+（加）、+（ 串联）、−（减） | 3 |
| ^（按位异或）、&（按位与）、|（按位或） | 4 |
| =、>、<、>=、<=、<>、!=、!>、!<（比较运算符） | 5 |
| not | 6 |
| and | 7 |
| all、any、between…and、in、like、or、some | 8 |
| =（赋值） | 9 |

#### 注意：同一行中运算符优先级相同，当表达式中含有优先级相同的多个运算符时，根据位置，二元运算符遵从从左到右的顺序执行，一元运算符遵从从右到左的顺序执行。

P120

【例7- 7】 计算各教师的各门、各班的课酬，其中，课酬=学时\*课酬标准，课酬标准分别为：教授150元/学时，副教授120元/学时，讲师100元/学时，助教60元/学时。

use jxgl

update 授课

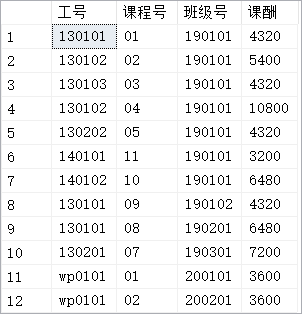
set 课酬 = 学时\* case

when 职称='教授' then 150

when 职称='副教授' then 120

when 职称='讲师' then 100

else 60

 end

from 教师,课程,授课

where 教师.工号=授课.工号

and 课程.课程号=授课.课程号

go

select \* from 授课

运行结果如图7-2所示。

P124页： 由于raiserror已被淘汰，不再建议使用。

#### 8．抛出异常语句

语法格式如下。

throw [{error\_number|@local\_variable},{message|@local\_variable},

{state|@local\_variable}][;]

功能：（在try…catch语句块的catch模块中）抛出异常处理信息。

说明：

（1）error\_number|@local\_variable：指定异常消息的标识号，可取值50000~2147483647之间的常量或变量，其数据类型为int；

（2）message|@local\_variable：指定异常消息的描述文本，可取字符串长度最大值为2048，其数据类型为nvarchar(n)。

（3）state|@local\_variable：指定异常消息的状态码，可取值0~255之间的常量或变量，其数据类型为tinyint。

【例7-11】在向“选修”表添加数据时，使用抛出异常语句来处理插入语句的插入异常信息。

use jxgl

go

begin try

insert into 选修(成绩编码,学号,课程号,成绩)values(20,'19020104','02',99)

end try

begin catch

throw 51000,'成绩编码列上有identity约束，不能人为插入值',1

end catch

运行结果如图7-3所示。



图7-3 例7-11运行结果

P173

1．在定义存储过程时，下面说法中不正确的是（ ）。

A．output关键词用于指定参数为输入参数类型

B． 如果定义了默认值，执行存储过程时可以不提供实参

C．varying用于指定作为输出参数支持的结果集，且仅适用于定义cursor输出参数

D．不推荐使用sp\_为前缀来创建用户自定义存储过程，因为sp\_前缀是用来命名系统存储过程的

2．关于存储过程，下面说法中错误的是（ ）。

A．存储过程可用于实施企业业务规则 B．存储过程由数据库服务器自动执行

C．存储过程可以使用游标 D．存储过程可以有输入输出参数

3．下列选项中不属于存储过程的优点的是（ ）。

A．增强代码的重用性和共享性

B．可以加快运行速度，减少网络流量

C．阻挡没有权限的用户间接存取数据库，从而保证数据的安全性

D．以使相关的动作在一起发生，从而可以维护数据库的完整性

4．以下触发器在对[表1]进行（ ）操作时触发。

create trigger abc on 表1

for insert,update,delete

as

……

A．只是更新 B．更新、插入、删除

C．只是删除 D．只是插入

P216页 （原因：sql\_loginB\_U在前面例中已经加入到db\_owner角色中。故修改如下）

【例11-16】 创建架构user\_schema\_sql，并指定sql\_loginA\_U为架构所有者。

use jxgl

go

create schema user\_schema\_sql authorization sql\_loginA\_U

运行结果如图11-38所示。

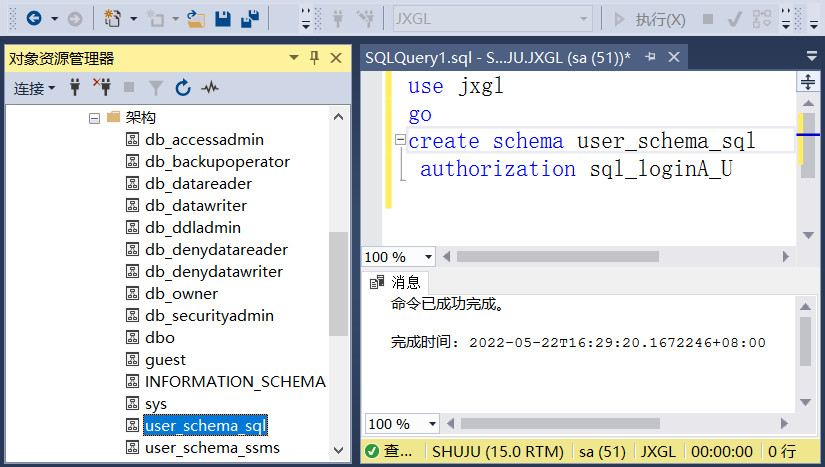


图11-38 sql\_loginA\_U的架构

P184

（3）在左侧“选择页”列表框中单击“备份选项”选项卡，在右侧为“备份数据库”设置“备份选项”选项卡信息，如图10-13所示。

P220

打开“数据库属性-JXGL”窗口，在“选择页”列表框中单击“权限”选项卡，如图11-52所示，在“用户或角色”列表框中选择用户或角色，并在“显式”列表框中根据需要设置该用户或角色的语句权限（授予/拒绝）。设置完所有用户或角色的语句权限后，单击“确定”按钮，返回SSMS，完成所有用户或角色的语句权限操作。

P208

### 11.4.1 角色类型概述

按照主体适用性划分，角色可分为3种类型：服务器角色、数据库角色和应用程序角色。

#### 1．服务器角色

服务器角色是操作服务器实例权限的分类描述和登录账户的逻辑分组。服务器角色在服务器级别定义并存储于每个服务器实例中。服务器角色又分为固定服务器角色和用户自定义服务器角色两类。

#### 2．数据库角色

数据库角色是操作数据库对象权限的分类描述和数据库用户的逻辑分组。数据库角色在数据库级别定义并存储于每个数据库中。数据库角色又分为固定数据库角色和用户自定义数据库角色两类。

（2）用户自定义数据库角色

用户自定义数据库角色是指由用户创建并定义权限的数据库角色，以便于集中管理同等操作权限的数据库用户。

#### 3．应用程序角色

应用程序角色是操作特定应用程序的权限描述。应用程序角色是不含任何成员的数据库（级别）主体，只允许通过特定应用程序连接的数据库用户访问特定数据。当数据库用户使用应用程序角色进行连接时，便自动放弃了所有数据库角色的权限。

P214，下面内容如涉及到版本结构，可以不修改

#### 注意：视图节点下出现名为dbo.sql\_loginB\_V的视图。验证了数据库用户sql\_loginB\_U获得了数据库角色db\_owner授予其创建视图的权限。

#### 3．使用T-SQL语句删除用户自定义数据库角色

格式：drop role <用户自定义的数据库角色名>

功能：删除指定名称的用户自定义数据库角色。

#### 注意：拥有架构的数据库角色不能被删除，拥有数据库用户的数据库角色不能被删除。

【例11-14】 使用T-SQL语句删除用户自定义数据库角色user\_role\_ssms（关联【例11-10】）。

use jxgl

go --删除数据库角色之前，需转移此数据库角色拥有的架构

alter authorization on schema::db\_accessadmin to db\_accessadmin

go --删除数据库角色之前，需删除此数据库角色中的成员

alter role user\_role\_ssms drop member win\_regA\_U

go

if exists(select \* from sys.database\_principals where name='user\_role\_ssms')

drop role user\_role\_ssms

#### 4．使用T-SQL语句修改应用程序角色

格式：alter application role <应用程序角色名> {with

[name=应用程序角色新名][ [, ] password='新密码'] [, default\_schema=新架构名]}

功能：修改应用程序角色的名称、密码或者默认架构的名称。

#### 5．使用T-SQL语句删除应用程序角色

格式：drop application role <应用程序角色名>

功能：删除指定名称的应用程序角色。

## 11.5 架构

架构是数据库对象的命名空间和数据库（级别）主体的操作平台。架构所有者拥有对架构下对象的所有权。一个数据库（级别）主体可以不拥有架构，也可以拥有多个架构。架构所有者可以将架构使用权转让给其他数据库（级别）主体，但始终保留拥有架构下对象的控制（control）权限。

### 11.5.1 架构概述

从管理者角度来看，架构可以看做是数据库对象管理的逻辑单位。架构下每个对象的完全限定名都是唯一的，其命名形式遵循服务器名.数据库名.架构名.对象名的格式。在每个数据库中，系统都提供了一系列内置架构，其中，有四个特殊的架构：dbo、guest、INFORMATION\_SCHEMA和sys，如图11-34所示。

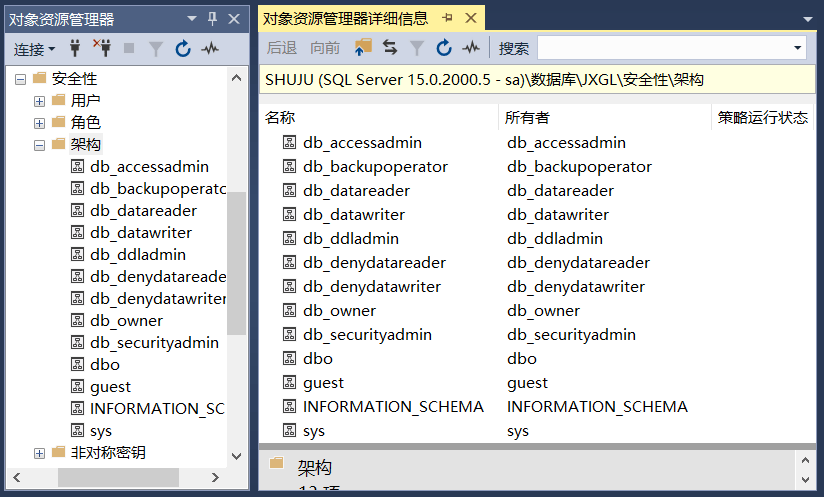


图11-34 系统内置架构名

（1）dbo是每个数据库的默认架构，其拥有者是数据库用户dbo；

（2）guest是访客的默认架构，其拥有者是数据库用户guest；

（3）INFORMATION\_SCHEMA是数据库引擎内部的架构，用户不能删除和修改。

（4）sys是系统对象（系统元数据、视图、函数）的默认架构；

### 11.5.2 创建架构

架构的创建有两种方式：使用SSMS和使用T-SQL语句。

#### 1. 使用SSMS创建架构

【例11-15】 在数据库JXGL中，使用SSMS创建架构user\_schema\_ssms，所有者是win\_regB\_U，并授权win\_regA\_U拥有访问架构的权限。

（1）在“对象资源管理器”窗格中展开数据JXGL的“安全性”节点，右击“架构”节点，在弹出的快捷菜单中执行“新建架构”命令，打开“架构-新建”窗口的“常规”选项卡，输入架构名称user\_schema\_ssms和架构所有者win\_regB\_U，如图11-35所示。

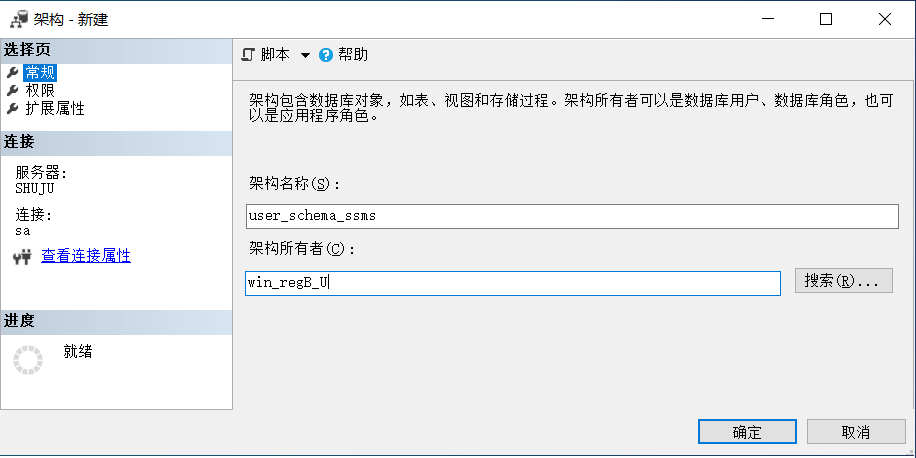


图11-35 “架构-新建”窗口的“常规”选项卡

（2）在左侧“选择页”列表框中选择“权限”选项卡，打开“权限”界面，在“用户或角色”栏中选择数据库用户win\_regA\_U，在“显示”列表框栏中显示“插入、更新、选择和删除”权限，如图11-36所示。

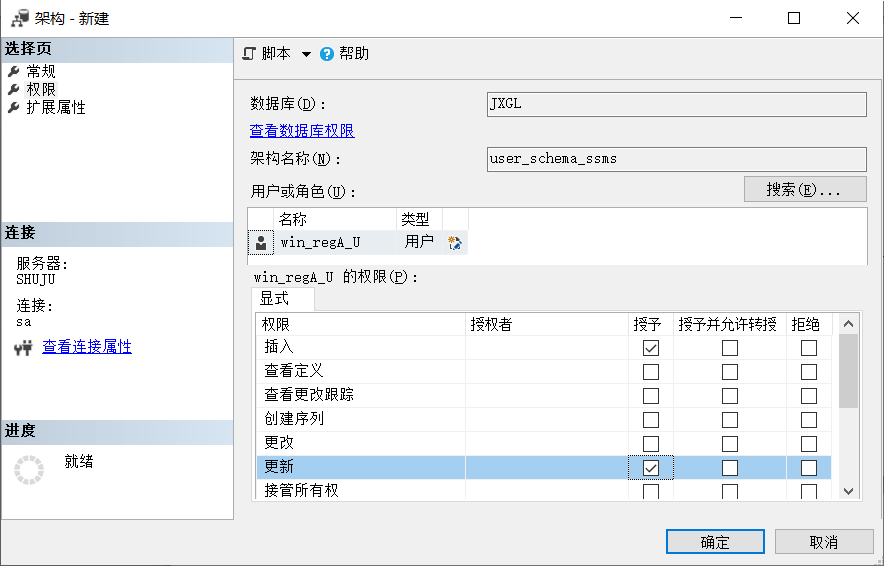


图11-36 “权限”选项卡

（3）单击“确定”按钮，返回SSMS，完成架构的创建以及成员权限的授予。

#### 注意：当使用win\_regB（或win\_regA）登录数据库服务器时，由于没有安全对象移入到架构user\_schema\_ssms中，因此，只可以查看数据库JXGL，不可以查看其中的各种数据库对象，如图11-37所示。

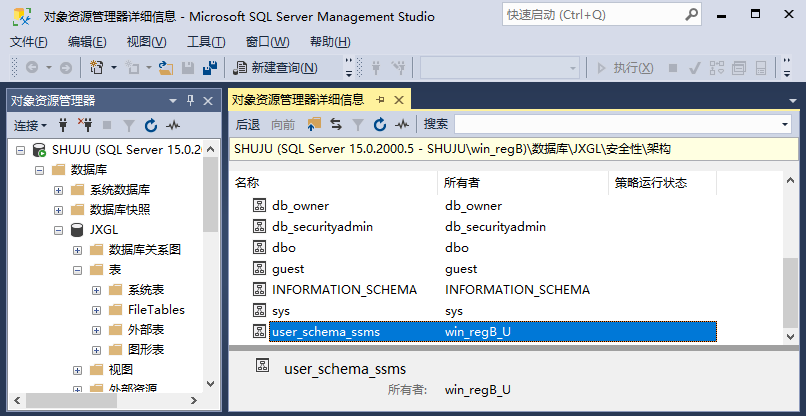


图11-37 win\_regB\_U的架构

#### 2. 使用T-SQL语句创建架构

格式：create schema 架构名 [authorization <数据库角色|数据库用户>]

功能：创建指定名称的用户自定义架构。

说明：authorization指定架构所有者，缺省为当前数据库用户。

#### 注意：多个数据库用户可以通过共享数据库角色或Windows组来拥有同一个架构。

【例11-16】 在数据库JXGL中，创建架构user\_schema\_sql，并指定数据库用户sql\_loginA\_U为架构所有者。

use jxgl

go

create schema user\_schema\_sql authorization sql\_loginA\_U

运行结果如图11-38所示。

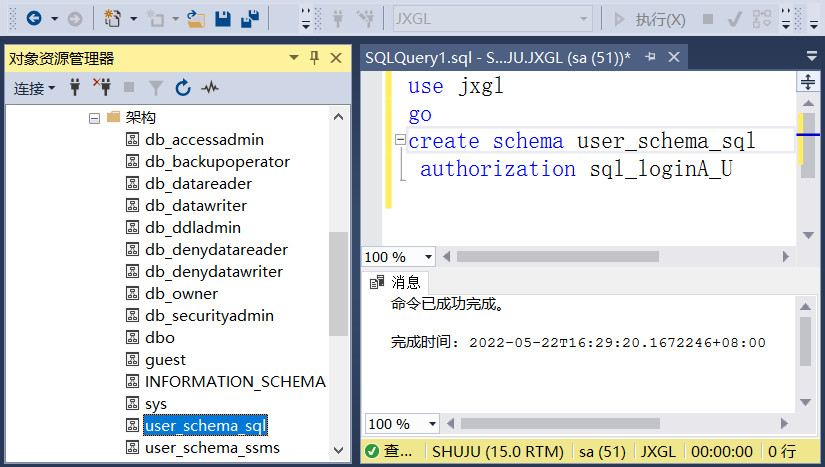


图11-38 sql\_loginA\_U的架构

### 11.5.3 管理架构

架构的管理内容很多，包括安全对象的移入、架构权限的授予、架构所有权的转移和删除架构。

1．安全对象的移入

格式：alter schema 架构名 transfer [<实体类型名>::]安全对象名

功能：将安全对象移入到指定名称的架构中。

说明：

（1）架构名：指定安全对象要移入的架构名称；

（2）实体类型名：指定安全对象的所有者的实体类型，可取值object、type或xml schema collection，默认为object。

（3）安全对象名：指定要移入当前架构范围内的安全对象名称。

【例11-17】 将数据库JXGL中的安全对象dbo.学生移入架构user\_schema\_sql中。

use jxgl

alter schema user\_schema\_sql transfer object::dbo.学生

运行结果如图11-39所示。

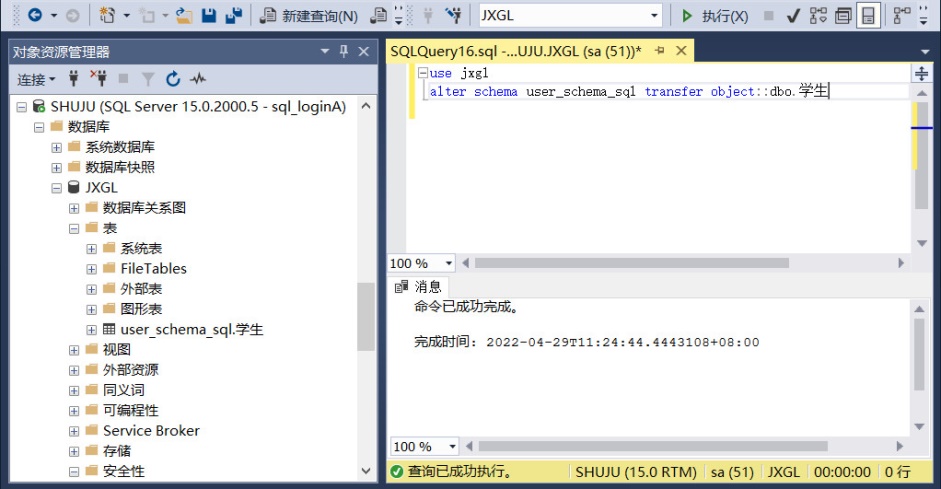


图11-39 sql\_loginA\_U拥有架构的安全对象

2．架构权限的授予

格式：grant {架构权限}[,...n] on schema::架构名 to <数据库主体>[,...n] [with grant option][as组|角色]

功能：将架构权限授予指定名称的数据库主体。

【例11-18】 在数据库JXGL中，验证数据库用户sql\_loginC\_U（需要事先创建）获取架构user\_schema\_sql下安全对象授予select权限的变化（关联【例11-16】和【例11-17】）。

操作步骤：

步骤①：以sa（或者等价sa的登录账户）登录数据库服务器，输入并执行以下代码：

use jxgl

go

create login sql\_loginC with password ='test'

go

create user sql\_loginC\_U for login sql\_loginC

步骤②：以sql\_loginC登录数据库服务器，输入并执行以下代码：

use jxgl

go

select \* from user\_schema\_sql.学生

运行结果如图11-40所示。验证了数据库用户sql\_loginC\_U没有获得架构user\_schema\_sql下安全对象的选择（select）权限。

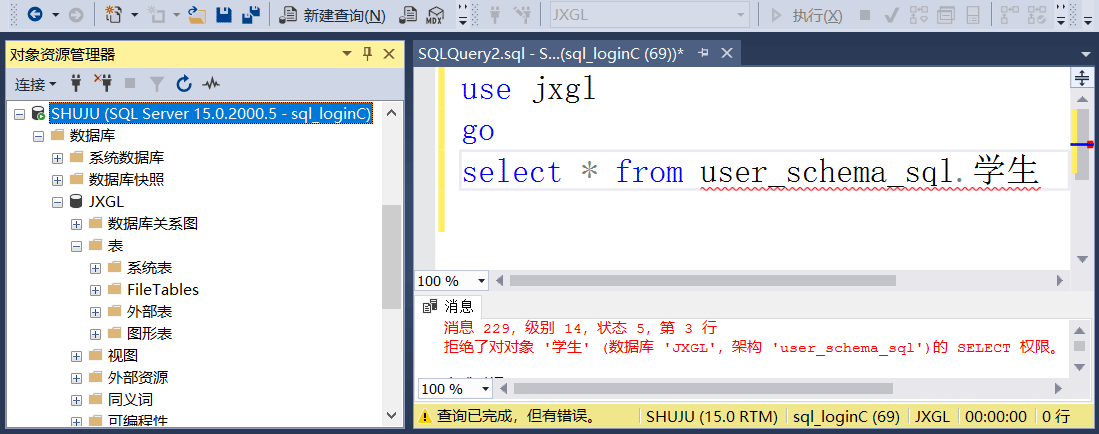


图11-40 sql\_loginC\_U没有架构的select权限

步骤③：以sa（或者等价sa的登录账户）登录数据库服务器，输入并执行以下代码：

use jxgl

go

grant *select* on schema::user\_schema\_sql to sql\_loginC\_U

运行结果如图11-41所示。验证了数据库用户sql\_loginC\_U获得了架构所有者授予的选择（select）权限。

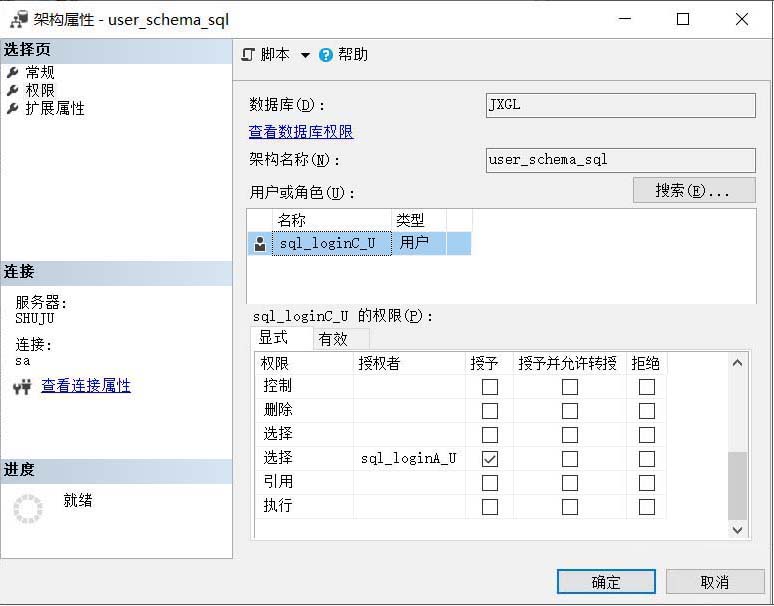


图11-41 sql\_loginC\_U获取了架构中select权限

步骤④：以sql\_loginC登录数据库服务器，输入并执行以下代码：

use jxgl

go

select \* from user\_schema\_sql.学生

运行结果如图11-42所示。验证了数据库用户sql\_loginC\_U成功执行架构user\_schema\_sql下安全对象的选择（select）权限。

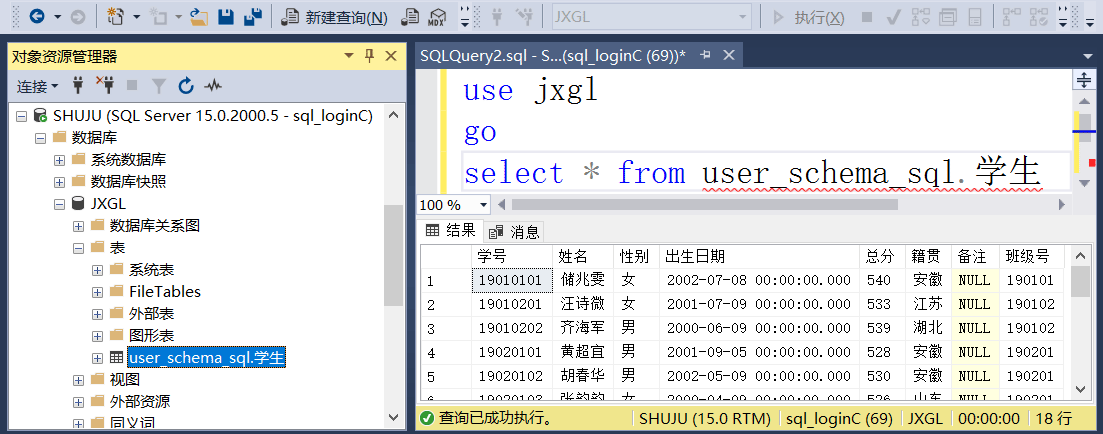


图11-42 sql\_loginC\_U获取了架构中select权限

3．架构所有权的转移

格式：alter authorization on schema::架构名to 数据库主体

功能：变更架构的所有者。

#### 注意：当数据库主体变更为dbo时，则表示撤销当前数据库主体对架构的所有权。

4．删除架构

格式：drop schema 架构

功能：删除指定名称的架构。

#### 注意：删除架构不影响架构所有者继续存在，包含安全对象的架构不能被删除。

【例11-19】 删除架构user\_schema\_sql（关联【例11-17】）。

use jxgl

go

alter schema dbo transfer object::user\_schema\_sql.学生

go

drop schema user\_schema\_sql

### P230-P231 严重错误

### 12.1.2 事务管理模式

根据set implicit\_transactions的取值（off|on）以及是否使用事务控制语句，DBMS中的事务管理模式分类3类：自动提交事务模式、显式事务模式和隐式事务模式。

#### 1．自动提交事务模式

……

……

……

【例12-1】 使用自动提交事务模式验证语法错误情况下批处理中语句的执行结果。

use jxgl

go

insert into 课程(课程号,课程名称) values('0a','劳动教育')

insert into 课程(课程号,课程名称) values('0b','廉洁教育')

insert into 课程(课程号,课程名称) vales('0c','学科前沿') --语法错误

go

select \* from 课程 where 课程号 in('0a','0b','0c') --无返回行

运行结果：第2个批处理中的第3个insert语句出现了编译错误，阻止了所有insert语句的执行，因此查询结果为空。

#### 2．显式事务模式

显式事务模式（set implicit\_transactions off）是由事务控制语句定义的事务管理模式。在显式事务模式下，每个事务必须由begin transaction语句开启事务（否则就是自动提交事务模式），并由commit或rollback语句结束事务。

请删除原文中的的【例12-1】

P236 原文中“串行”是个严重错误。

#### 注意：事物的并行调度提高了系统的事物吞吐量（单位时间内执行的事物条数）和资源利用率，改善短事物的响应时间，但可能会破坏数据的一致性。

#### 3．可串行化调度

P238

### 12.2.2 封锁协议

……

……

……

1.一级封锁协议

……

……

表12-11 一级协议与防止丢失修改

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 时刻 | 事务甲 | 事务乙 | 时刻 | 事务甲 | 事务乙 |
| t0 | 获准Xlock(A) |  | t5 |  | 获准Xlock(A) |
| t1 | Read(A)=50 |  | t6 |  | Read(A)=47 |
| t2 | A=A-3 | 申请Xlock(A) | t7 |  | A=A-2 |
| t3 | Write(A)=47 | wait | t8 |  | Write(A)=45 |
| t4 | Unlock(A) | wait | t9 |  | Unlock(A) |

2.二级封锁协议

……

……

……

12-12 二级封锁协议与解决读“脏”数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 时刻 | 事务甲 | 事务乙 | 时刻 | 事务甲 | 事务乙 |
| t0 | 获准Xlock(A) |  | t4 | Rollback | wait |
| t1 | Read(A)=50 |  | t5 | Unlock(A) | wait |
| t2 | A=A-3 |  | t6 |  | 获准Slock(A) |
| t3 | Write(A)=47 | 申请Slock(A) | t7 |  | Read(A)=50 |

3.三级封锁协议

……

……

……

表12-13 三级封锁协议与解决不可重复读

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 时刻 | 事务甲 | 事务乙 | 时刻 | 事务甲 | 事务乙 |
| t0 | 获准Slock(A) |  | t5 |  | 获准Xlock(A) |
| t1 | Read(A)=50 |  | t6 |  | Read(A)=50 |
| t2 |  | 申请Xlock(A) | t7 |  | A=A+50 |
| t3 | Read(A)=50 | wait | t8 |  | Write(A)=100 |
| t4 | Unlock(A) | wait | t9 |  | Unlock(A) X |

1. 如果假定X是人，Z则是人的摆拍映像，像集就是一个个特定人的各自不同摆拍映像的集合。 [↑](#footnote-ref-1)