第 10 章

存储过程与存储函数

存储过程与存储函数是 MySQL 支持的过程式数据库对象，能够将复杂的 SQL 逻辑封装在一起，应用程序无须关注存储过程和存储函数内部复杂的 SQL 逻辑，只需要简单地调用存储过程和存储函数即可。使用存储过程和存储函数可以提高数据库的处理速度，提高数据库编程的灵活性。

学习目标

- 了解什么是存储过程与存储函数。
- 了解存储程序的类型。
- 理解存储过程与存储函数的作用。
- 掌握存储过程与存储函数的创建和管理。

10.1 存 储 过 程

存储过程是存储在数据库目录中的一段声明式 SQL 语句，它可以被触发器、其他存储过程及应用调用，调用自身的存储过程称为递归存储过程。MySQL 的存储过程有利有弊，在开发过程中，要根据自己的业务需求决定是否使用存储过程。

10.1.1 什么是存储过程

在大型数据库系统中，存储过程是一组为了完成特定功能的 SQL 语句集。一个存储过程是一个可编程的函数，它在数据库中创建并保存，一般由 SQL 语句和一些特殊的控制结构组成。使用存储过程不仅可以提高数据库的访问效率，同时也可以提高数据库使用的安全性。

MySQL 5.0 版本以前并不支持存储过程，这使 MySQL 在应用上大打折扣。MySQL 从 5.0 版本开始支持存储过程，既提高了数据库的处理速度，同时也提高了数据库编程的灵活性。

10.1.2　存储程序的类型

存放在 MySQL 服务器端，供重复使用的对象叫作存储程序。存储程序分为以下 4 种：

1. 存储过程（stored procedure）

存储过程不直接返回一个计算结果，但可以用来完成一般的运算或是生成一个结果集并传递回客户端。一条 SQL 语句如果比作一行 java 代码，存储过程就相当于一个 java 方法，可以包含许多 SQL 语句，能进行更复杂的操作。

2. 存储函数（stored function）

存储函数返回一个计算结果，该结果可以用在表达式里。就相当于自定义 MySQL 函数一样，它的作用和 MySQL 函数类似，只不过需要用户自己去定义。

3. 触发器（trigger）

触发器与数据表相关联，当那个数据表被 INSERT、DELETE 或 UPDATE 语句修改时，触发器将自动执行。如果表关联了触发器，当表数据有修改操作时，触发器将自动执行，至于做什么是自定义的。

4. 事件（event）

事件根据时间表在预定时刻自动执行。例如，可以自己设定一个开始时间，然后让它每隔指定的时间段重复做某些事情。

10.1.3　存储过程的作用

MySQL 存储过程具有以下 5 个作用：

（1）存储过程的使用，提高了程序设计的灵活性。存储过程可以使用流程控制语句组织程序结构，方便实现结构较复杂的程序的编写，使设计过程具有很强的灵活性。

（2）存储过程把一组功能代码作为单位组件。一旦被创建，存储过程作为一个整体，可以被其他程序多次反复调用。

（3）MySQL 存储过程是按需编译的。在编译存储过程之后，MySQL 将其放入缓存中。MySQL 为每个连接维护自己的存储过程高速缓存。如果应用程序在单个连接中多次使用存储过程，则使用编译版本，否则存储过程的工作方式类似于查询。

（4）存储过程有助于减少应用程序和数据库服务器之间的流量。因为应用程序不必发送多个冗长的 SQL 语句，只发送存储过程中的名称和参数即可。

（5）存储的程序是安全的。数据库管理员可以向访问数据库中存储过程的应用程序授予适当的权限，但不向基础数据库表提供任何权限。

10.1.4 创建存储过程

1. 利用 CREATE PROCEDURE 语句创建存储过程

在 MySQL 中，创建存储过程的语法格式如下：

```
CREATE PROCEDURE sp_name([proc_parameter[,…]])
[characteristic…]
routine_body
```

参数说明：

- CREATE PROCEDURE：创建存储过程的关键字。
- sp_name：存储过程的名称。建立这个名称时，要避免和 MySQL 内置函数的名称相同。
- proc_parameter：存储过程的参数列表。格式如下：

```
[IN|OUT|INOUT]param_name type
```

  - IN 表示输入参数。
  - OUT 表示输出参数。
  - INOUT 表示既可以输入参数也可以输出参数。
  - Param_name 表示参数名称。
  - type 表示参数的类型。

参数的命名要避免和数据表的字段名相同。

- characteristic：存储过程的特性。格式如下：

```
LANGUAGE SQL | [NOT] DETERMINISTIC | {CONTAINS SQL|NO SQL|READS SQL DATA|MO
DIFIES SQL DATA} | SQL SECURITY{DEFINER|INVOKER} | COMMENT 'string'
```

  - LANGUAGE SQL：说明 routine_body 部分是由 SQL 语句组成的，当前系统支持的语言为 SQL。SQL 是 LANGUAGE 特性的唯一值。
  - [NOT] DETERMINISTIC：指明存储过程执行的结果是否正确。默认为 NOT DETERMINISTIC。
  - {CONTAINS SQL|NO SQL|READS SQL DATA|MODIFIES SQL DATA}：CONTAINS SQL 表示存储程序不包含读或写数据的语句；NO SQL 表示存储程序不包含 SQL 语句；READS SQL DATA 表示存储程序包含读数据的语句；MODIFIES SQL DATA 表示存储程序包含写数据的语句。
  - SQL SECURITY{DEFINER|INVOKER}：指明谁有权限来执行该存储过程。DEFINER 表示只有定义者才能执行。INVOKER 表示拥有权限的调用者可以执行存储程序。
  - COMMENT ‘string’：注释信息，可以用来描述存储过程或函数。

- routine_body：表示存储过程的程序体，以 BEGIN 表示 SQL 代码的开始，以 END 表示 SQL 代码的结束。如果存储过程的程序体中仅有一条 SQL 语句，可以

省略 BEGIN 和 END 标志。

【例 10-1】创建一个存储过程 proc_stud，从数据库 studentgradeinfo 的 student 表中检索出所有民族为“汉族”的学生的学号、姓名、性别等相关信息。如图 10-1 所示。SQL 代码如下：

```
--打开 studentgradeinfo 数据库
USE studentgradeinfo;
--创建存储过程
CREATE PROCEDURE proc_stud()
BEGIN
   SELECT StudentId, StudentName,StudentSex,StudentClass,StudentDepartmen
t,
   StudentNation,StudentPolitics,StudentBirthday FROM student
   WHERE StudentNation LIKE '%汉族%' ORDER BY StudentId;
END;
CALL proc_stud();
```

执行存储过程 proc_stud，返回所有民族是“汉族”的学生信息，执行结果如图 10-1 所示。

```
1  --打开studentgradeinfo数据库
2  USE studentgradeinfo;
3  --创建存储过程
4  CREATE PROCEDURE proc_stud()
5  BEGIN
6     SELECT StudentId, StudentName,StudentSex,StudentClass,StudentDepartment,
7     StudentNation,StudentPolitics,StudentBirthday FROM student
8     WHERE StudentNation LIKE '%汉族%' ORDER BY StudentId;
9  END;
10 CALL proc_stud();
11 执行存储过程proc_stud，返回所有民族是"汉族"的学生信息
```

信息　结果1　概况　状态

StudentId	StudentName	StudentSex	StudentClass	StudentDepartment	StudentNation	StudentPolitics	StudentBirthday
1001001	赵明亮	男	计算机2001	信息工程	汉族	团员	2001-02-15 00:00:00
1001002	钱多多	男	计算机2001	信息工程学院	汉族	党员	2001-08-25 00:00:00
1002001	李静	女	网络2002	信息工程学院	汉族	团员	2000-01-20 00:00:00
1002005	魏金木	男	网络2002	信息工程学院	汉族	群众	2002-08-28 00:00:00
2002001	韦谨言	男	商务2002	工商管理学院	汉族	群众	2001-05-16 00:00:00
2002002	黄慧	女	商务2002	工商管理学院	汉族	群众	2001-07-07 00:00:00
3001001	李运国	男	动漫2001	艺术学院	汉族	群众	2001-11-25 00:00:00
3001002	张轩宇	男	动漫2001	艺术学院	汉族	群众	2001-06-06 00:00:00
4001001	李强	男	机械2001	机械工程学院	汉族	团员	2001-09-10 00:00:00
5001001	李佳欣	女	汽修2001	交通工程学院	汉族	党员	2002-01-12 00:00:00

图 10-1　执行存储过程 proc_stud

2. 利用 Navicat 创建存储过程

利用 Navicat 创建存储过程更简单、方便，操作步骤如下：

（1）在 Navicat 中连接 MySQL 服务器。

（2）分以下两种方法创建存储过程。

方法 1：在 localhost_3306→studentgradeinfo 下，单击“新建函数”按钮，如图 10-2 所示。

方法 2：用鼠标右击 studentgradeinfo 下的“函数”，在弹出的快捷菜单中选择“新建函数”命令。

图 10-2　新建函数

（3）打开“函数向导”窗口，选择要创建的例程类型，选择“过程”，单击“下一步”按钮，如图 10-3 所示。

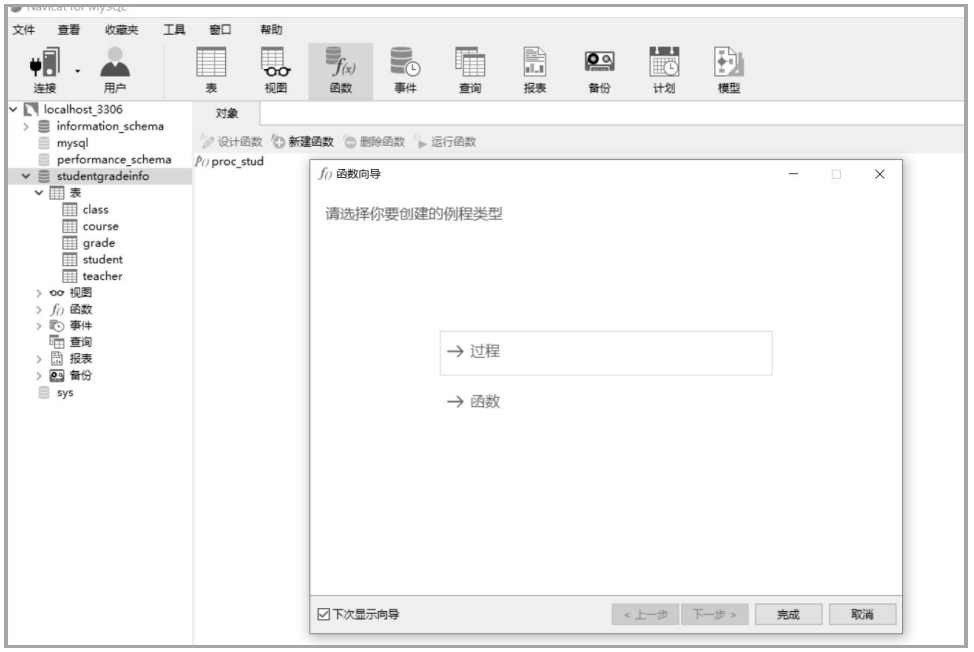

图 10-3　选择例程类型

（4）进入参数设置界面，其中“模式”表示参数的类型（IN、OUT、INOUT），“名”表示参数的名称，“类型”表示该参数的数据类型。如果参数不止一个，可以单击左下方的“＋”按钮添加；如果要删除参数，可以单击左下方的“－”按钮。“↑”按钮和“↓”按钮可以使光标在各个参数之间转移，如图 10-4 所示。

图 10- 4　设置参数

（5）参数设置完成后，单击“完成”按钮，进入下一步操作。此时用户就可以在 BEGIN 和 END 之间输入 SQL 语句了。输入完成后，单击窗口上方的“保存”按钮，如图 10-5 所示。

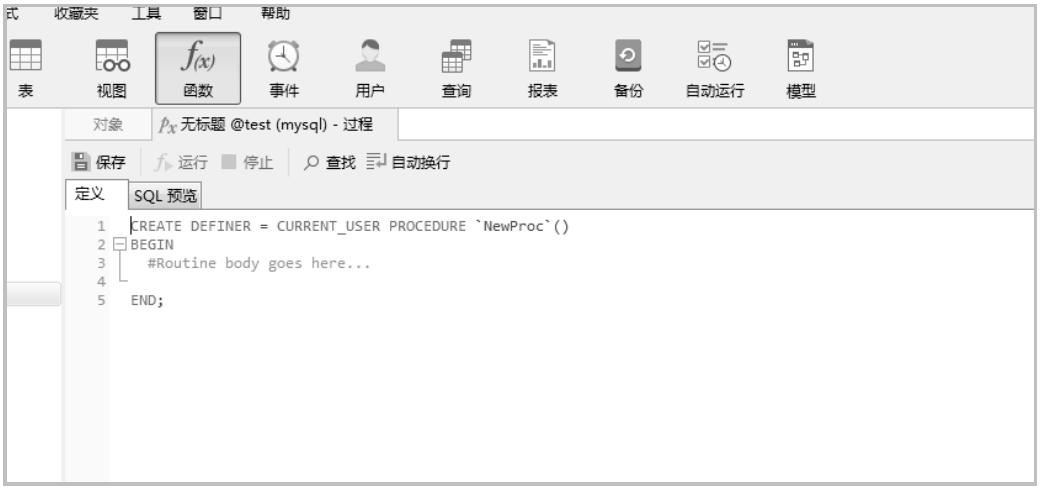

图 10-5　输入 SQL 语句

10.1.5　管理存储过程

1．查看存储过程

存储过程和函数的信息存储在 information_schema 数据库下的 Routines 表中。可以通过查询该表的记录来查询存储过程的信息。其基本语法格式如下：

```
select name from mysql.proc where db='数据库名';
select routine_name from information_schema.Routines
```

```
where routine_ schema = '数据库名';
```

【例 10-2】查看 studentgradeinfo 数据库内存储过程信息。SQL 语句如下：

```
USE studentgradeinfo;
select * from mysql.proc where db='studentgradeinfo';
```

在 MySQL 中，存储过程创建以后，用户也可以使用 show status 语句或 show create 语句来查看存储过程的名称和详细信息，以及存储过程的定义信息。其基本语法格式如下：

显示数据库内存储过程的名称和详细信息：

```
SHOW PROCEDURE status [LIKE 'pattern'];
```

PROCEDURE 表示查询存储函数；LIKE‘pattern’用来匹配存储过程的名称。

查看存储过程的定义语句等信息：

```
SHOW CREATE PROCEDURE 数据库.存储过程名
```

【例 10-3】查看存储过程 p_student 的定义。SQL 语句如下：

```
USE studentgradeinfo;
SHOW CREATE PROCEDURE p_student\G ;
```

2. 修改存储过程

在 MySQL 中修改存储过程通过 ALTER PROCEDURE 完成。其基本语法格式如下：

```
ALTER PROCEDURE proc_name [characteristic... ]
```

参数 proc_name 表示存储过程或函数的名称；Characteristic 参数指定存储过程或函数的特性。

【例 10-4】修改存储过程 p_student 的定义，将读写权限改为 MODIFIES SQL DATA，并指明调用者可以执行。SQL 语句如下：

```
ALTER PROCEDURE p_student
  MODIFIES SQL DATA
  SQL SECURITY INVOKER;
```

3. 删除存储过程

在 MySQL 中删除存储过程，可以通过 DROP PROCEDURE 语句完成。其基本语法格式如下：

```
DROP PROCEDURE[IF EXISTS] proc_name;
```

其中，关键字 DROP PROCEDURE 用来实现删除存储过程；参数 proc_name 表示所要删除的存储过程名称。执行删除时如果存储过程不存在，使用 IF EXISTS 语句即可防止发生错误。

【例 10-5】删除 studentgradeinfo 数据库中的存储过程 proc_stud。SQL 语句如下：

```
USE studentgradeinfo;
DROP PROCEDURE proc_stud
```

10.2　存 储 函 数

在 MySQL 中，存在一种与存储过程十分相似的过程式数据库对象——存储函数。它与存储过程一样，都是由 SQL 语句和过程式语句组成的代码片段，并且可以被应用程序和其他 SQL 语句调用。

10.2.1　MySQL 常用函数

（1）字符串函数及功能，如表 10-1 所示。

表 10-1　字符串函数及功能

函 数 名 称	功 能 描 述
ASCII（char）	返回字符的 ASCII 码值
CONCAT(s1,s2...，sn)	将字符串 s1,s2...，sn 连接成一个字符串
LOWER(str)	将字符串转为小写字符
UPPER(str)	将字符串转为大写字符
LEFT(str, x)	返回字符串中最左边的 x 个字符
RIGHT(str, x)	返回字符串中最右边的 x 个字符
LENGTH(s)	返回字符串中的字符数
LTRIM(str)	从字符串中去掉开头的空格
RTRIM(str)	从字符串中去掉尾部的空格
TRIM(str)	去除字符串首部和尾部的所有空格
POSITION(substr,str)	返回子串 substr 在字符串 str 中第一次出现的位置
REVERSE(str)	返回颠倒字符串 str 后的结果
SRTCMP(s1,s2)	比较字符串 s1 和 s2 的大小，s1 比 s2 小时返回-1，s1 比 s2 大则返回 1，s1 等于 s2 时返回 0

（2）数学函数及功能，如表 10-2 所示。

表 10-2　数学函数及功能

函 数 名 称	功 能 描 述
ABS(x)	返回 x 的绝对值
BIN(x)	返回 x 的二进制值
CEILING(x)	返回大于等于 x 的最小整数值
EXP(x)	返回自然对数 e 的 x 次方
FLOOP(x)	返回小于等于 x 的最大整数值
LN(x)	返回 x 的自然对数
LOG(x,y)	返回 x 以 y 为底的对数
MOD(x,y)	返回 x / y 的余数

续表

函 数 名 称	功 能 描 述
PI()	返回圆周率的值
RAND( )	返回 0 到 1 的随机数
ROUND(x,y)	返回 x 四舍五入 y 位小数的值
SIGN(x)	返回 x 的符号。其中-1 代表负数，1 代表正数，0 代表 0
SQRT(x)	返回 x 的平方根

（3）日期和时间函数及功能，如表 10-3 所示。

表 10-3　日期和时间函数及功能

函 数 名 称	功 能 描 述
CURDATE()、CURTIME()	获取当前的系统日期或系统时间
NOW()	返回当前日期和时间值，格式为 YYYY-MM-DD HH:MM:SS
YEAR(date)	返回 date 对应的年份，范围是 1000～9999
MONTH(date)	返回 date 对应的月份，范围是 1～12
DAY(date)	返回 date 对应的天，范围是 1～31
WEEK (date)	返回 date 对应的周数，范围是 0～53
WEEKDAY(date)	返回 date 对应的工作日索引，0 表示周一，6 表示周日
DAYNAME(date)	返回 date 对应的工作日的英文名称，如 Sunday、Monday 等

10.2.2　存储过程与存储函数的联系与区别

存储过程和存储函数是事先经过编译并存储在数据库中的一段 SQL 语句的集合，调用存储过程和存储函数可以简化应用开发人员的很多工作，减少数据在数据库和应用服务器之间的传输，能够提高数据处理的效率。

存储过程和存储函数有如下相同点：

（1）存储过程和存储函数都是可重复执行操作的数据库的 SQL 语句的集合。

（2）存储过程和存储函数都是一次编译后缓存起来，下次使用就直接调用已经编译好的 sql 语句，减少网络交互，提高了数据处理的效率。

存储过程和存储函数有如下不同点：

（1）标识符不同，存储函数的标识符是 function，存储过程是 procedure。

（2）存储函数由于本身就要返回处理的结果，所以不需要输出参数，而存储过程则需要用输出参数返回处理结果。

（3）存储函数不需要使用 CALL 语句进行调用，而存储过程必须使用 CALL 语句进行调用。

（4）存储函数必须使用 RETURN 语句返回结果，存储过程不需要使用 RETURN 语句返回结果。

10.2.3　创建存储函数

1. 利用 CREATE FUNCTION 语句创建存储函数

在 MySQL 中创建存储函数的语法格式如下：

```
CREATE FUNCTION func_name ([func_parameter [,...]])
RETURNS type
[characteristic[,...]
Routine_body
```

参数说明：

- CREATE FUNCTION：创建存储函数的关键字。
- func_name：存储函数的名称。
- func_parameter：存储函数的参数列表，形式如 [IN| OUT| INOUT]param_name type。
- RETURNS type：指定返回值的数据类型。
- characteristic：可选项，指定存储函数的特性。
- Routine_body：SQL 代码内容。

2. 利用 Navicat 创建存储函数

利用 Navicat 创建存储函数与创建存储过程的方法相似，操作步骤如下：

（1）在 Navicat 中连接 MySQL 服务器。

（2）通过以下两种方法创建存储函数。

方法 1：在 localhost_3306→studentgradeinfo 下，单击窗格上方的“新建函数”按钮，如图 10-6 所示。

方法 2：用鼠标右击 studentgradeinfo 下的“函数”，在弹出的快捷菜单中选择“新建函数”命令。

图 10-6　新建函数

（3）打开“函数向导”窗口，选择要创建的例程类型，选择“函数”，单击“下一步”按钮，如图 10-7 所示。

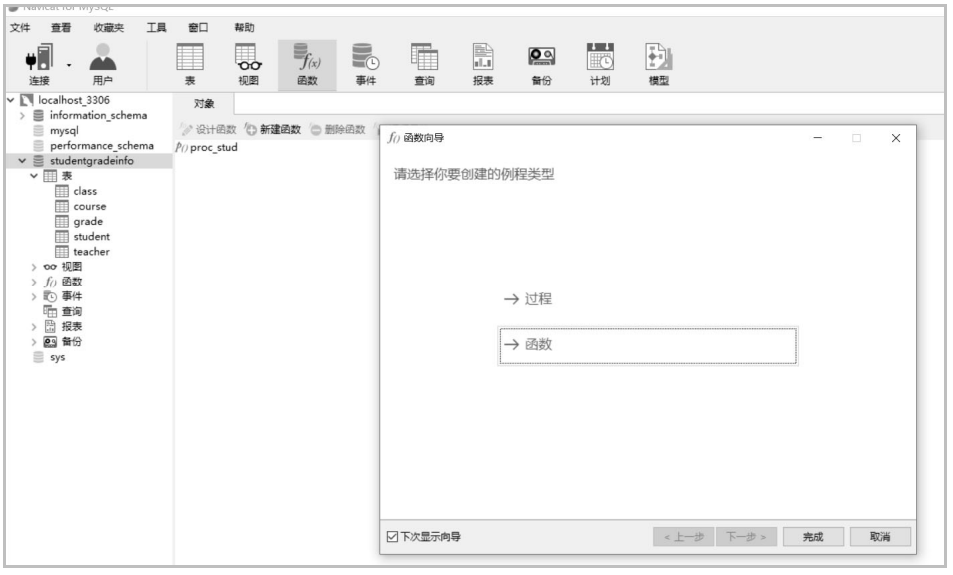

图 10-7　选择例程类型

（4）进入参数设置界面，其中“名”表示参数的名称，“类型”表示该参数的数据类型。各按钮的功能与创建存储过程相同，如图 10-8 所示。

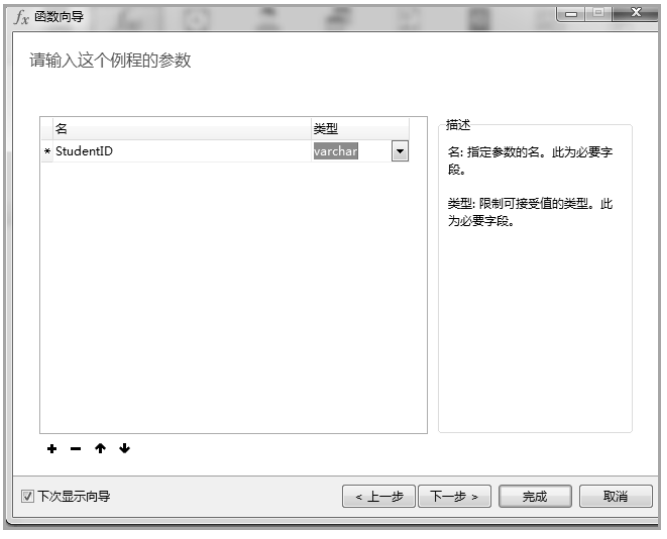

图 10-8　设置参数

（5）参数设置完后，单击“下一步”按钮，进入返回值设置界面，可以设置返回值类型、长度以及字符集等信息，如图 10-9 所示。

（6）设置完成后，单击“完成”按钮，进入定义窗口，输入存储函数代码。输入完成后，单击“保存”按钮，即可创建存储函数，如图 10-10 所示。

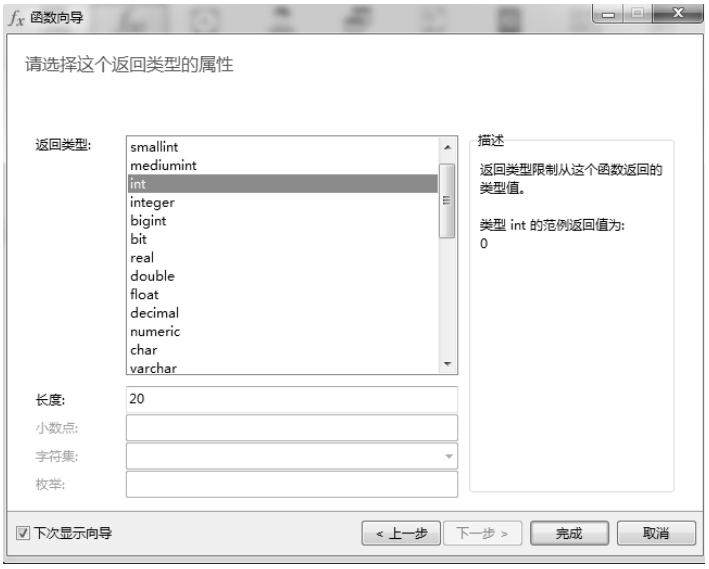

图 10-9　设置返回类型的属性

注意：在函数体开头处加上 READS SQL DATA 声明。

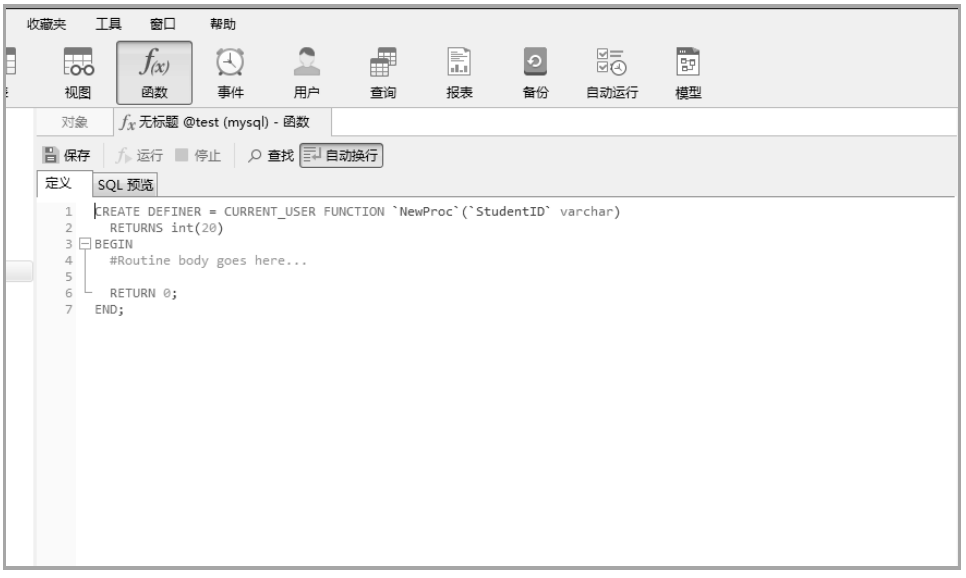

图 10-10　输入 SQL 语句

10.2.4　管理存储函数

1. 查看存储函数

和存储过程相同，存储函数被创建之后，也可以使用同样的方法进行查看。其语法格式如下：

```
select name from mysql.proc where db='数据库名';
select routine_name from information_schema.Routines where routine_ schema
 = '数据库名';
```

MySQL 存储了存储过程和存储函数的状态信息，用户也可以使用 show status 语句或 show create 语句来查看存储函数的名称、存储函数的详细信息以及存储函数的定义信息。其语法格式如下：

显示数据库内所有存储函数的名称和存储函数的详细信息：

```
SHOW FUNCTION status [LIKE 'pattern'];
```

FUNCTION 表示查询存储函数；LIKE‘pattern’用来匹配存储函数的名称。

查看指定存储函数的定义信息：

```
SHOW CREATE FUNCTION 数据库.存储函数名
```

2. 删除存储函数

在 MySQL 中删除存储函数，可以通过 DROP FUNCTION 语句来完成。其语法格式如下：

```
DROP FUNCTION[IF EXISTS] fn_name;
```

关键字 DROP FUNCTION 用来表示实现删除存储函数；参数 fn_name 表示所要删除的存储函数名称。

思政小课堂

不仅程序需要进行不同的选择，我们的人生也需要面对一个又一个的选择。例如，高中选文科还是理科，大学选哪所学校，专业该怎么选，未来是选择继续深造还是就业。每一次的选择都影响着我们的一生，在面临选择时，我们要以正确的价值观、人生观和世界观作为基石，从而做出适合自己的选择。不管如何选择，我们都要更好地去学习和生活，把自己的小我融入祖国的大我中，紧跟时代的脚步，将中国梦与我们个人的梦想紧密结合在一起，更好地实现自己的人生价值，早日实现中国梦！

10.3 总结与训练

本章主要介绍存储过程与存储函数，以 studentgradeinfo 数据库为例，从存储过程和存储函数的创建、查看、修改、删除等方面进行了学习。

实践任务：存储过程与函数的基本操作

1. 实践目的

（1）理解存储过程与存储函数的异同。

（2）掌握存储过程的创建、查看、修改、删除方法。

（3）掌握存储函数的创建、查看、删除方法。

2. 实践内容

（1）在 studentgradeinfo 数据库中创建一个名为 proc_teacher 的存储过程，该存储过程输出 teacher 表中所有 TercherDepartment 字段为“信息工程”的记录。

（2）查看 studentgradeinfo 数据库内存储过程的名称和详细信息。

（3）创建一个名为 func_name 的存储函数，返回“信息工程”的教师名单。

（4）查询名为 func_name 的存储函数的状态。

（5）简述存储过程与存储函数的异同。

13.6 运行环境描述

图书管理系统的开发和运行需要满足的硬件需求和必要的开发工具如下：

- 操作系统：WINDOWS 7 以上或 LINUX。
- 数据库：MySQL 5.6 以上。
- 数据库管理系统：Navicat Premium 12。

13.7 本 章 小 结

本章以开发图书管理系统为例，重点介绍了实现该系统的数据库分析设计的过程。通过本章内容的学习，读者不仅系统性地复习了 MySQL 的基础知识，而且对 MySQL 在实际项目中的应用和系统项目设计的过程进行了充分的了解。