数据库对象管理

1、用户及模式管理

DM7 数据库支持多模式结构，用户可以根据自己的需求创建不同的模式，方便对数据的分类，以及提供安全性保障。

模式是 DM 数据库的逻辑单位。一个模式可以包含许多数据库对象，同时每一个模式对应一个用户。通常每个用户都有一个同名的模式对应，即用户是该模式的拥有者。当使用用户名和密码连接到数据库上时，及可以访问同名模式的所有对象。

例 ：创建用户 USER01，密码为 123456789，通过 MANAGER 管理工具，可以看到已经自动生成 USER01 模式，通过 USER01 模式 管理 USER01 用户的所有对象。

1、 管理员用户 SYSDBA 创建用户 USER01 ，并授予 USER01 用户 DBA 权限

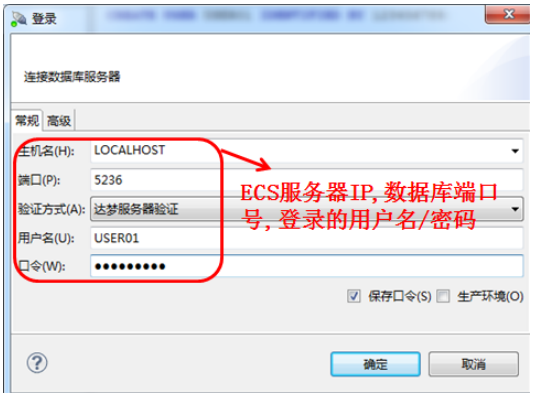
```
CREATE USER USER01 IDENTIFIED BY 123456789;
grant DBA to USER01;
```

2、 使用 USER01 用户创建表 TEST

```
CREATE TABLE TEST (ID INT ,NAME VARCHAR) ;
```

3、 通过 DM7 提供的 MANAGER 管理工具查看生成的 USER01 模式

1) 打开 MANAGER 管理工具，然后使用 USER01 用户名和密码登陆；



2) 点击左侧对象导航栏下的模式选项，已经自动生成 USER01 模式，创建的 TEST 表也已经存在于 USER01 模式下，则显示如下：

从图中可以看到 DM7 提供了表、外部表、视图、物化视图、存储过程等模式对象。

更多详细内容请参考 DM7 系统管理员手册或者 DM7_SQL 语言使用手册，或者联系在线客服。

2、模式对象管理

1) 表对象

可以使用 DM 管理工具或者 DDL 语句 CREATE TABLE 创建表。下面介绍用 DM 管理工具创建表。

例：使用 DM 管理工具，在 SYSDBA 模式下创建 DEPARTMENT 表：

① 展开**模式**：显示数据库中的所有模式

② 展开 **SYSDBA**：显示该模式下的对象，包括表、视图等

③ 在表上**右键**:显示菜单项

1. 点击**新建表**

2. 表名：键入 DEPARTMENT

3. 点击  图标添加列，输入以下信息：

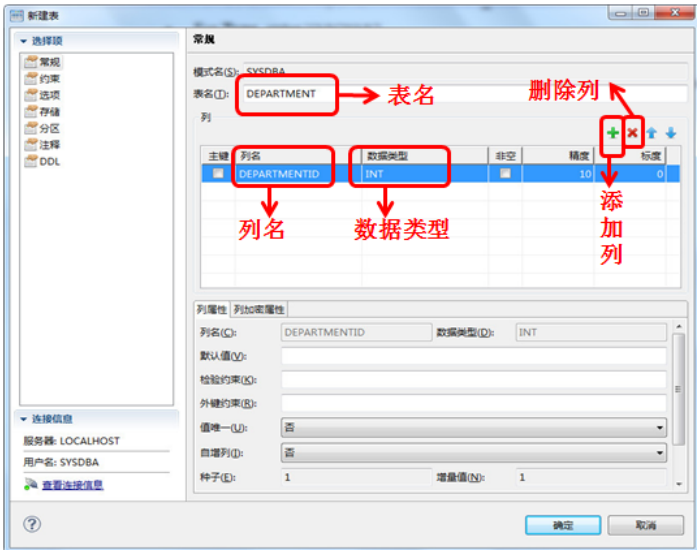
   列名：键入 DEPARTMENTID

   数据类型：选择 INT

   主键、非空、精度、标度：使用默认值

4. 点击**确定**：创建表 DEPARTMENT 成功。

5. 操作图



2) 索引对象

索引是与表相关的可选的结构，它能使对应于表的 SQL 语句执行得更快， DM7 索引能提供访问表的数据的更快路径，可以不用重写任何查询而使用索引，其结果与不使用索引是一样的，但速度更快。

可以使用 DM 管理工具或者 DDL 语句添加索引。下面介绍用 DM 管理工具添加索引。

例：使用 DM 管理工具，SYSDBA 模式下创建 EMPLOYEE 表并添加索引

1、 创建表的过程同表管理，此处省略

2、 添加索引

① 展开**模式**: 显示数据库中的所有模式

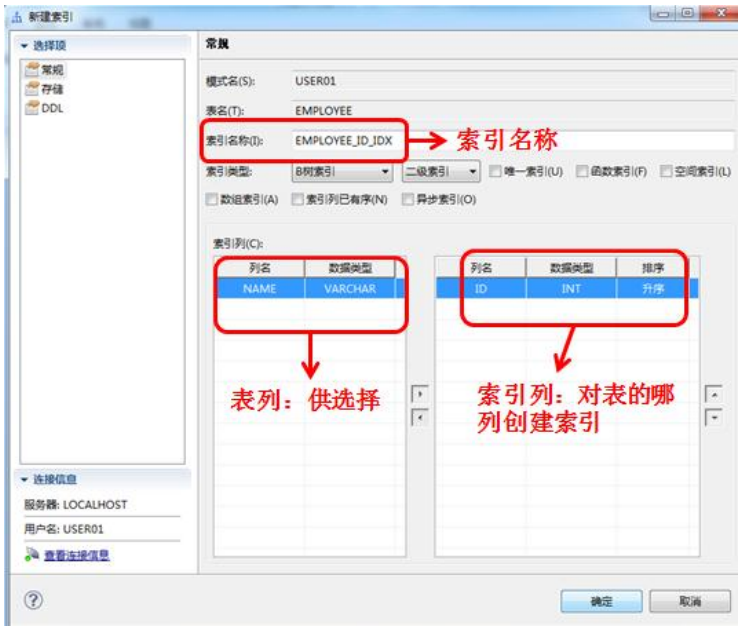
② 展开 **SYSDBA:** 显示该模式下的对象，包括表、视图等

③ 展开**表:** 显示 SYSDBA 下的表

④ 展开 **EMPLOYEE**：显示该表下的对象，包括列、键等

⑤ 在**索引**上右键：显示菜单项

1. 点击**新建索引**
2. 索引名称: 键入 EMPLOYEE_ID_IDX
3. 索引列：选中 ID，点击 
4. 点击**确定:** 创建索引 EMPLOYEE_ID_IDX 成功
5. 操作截图



3) 视图对象

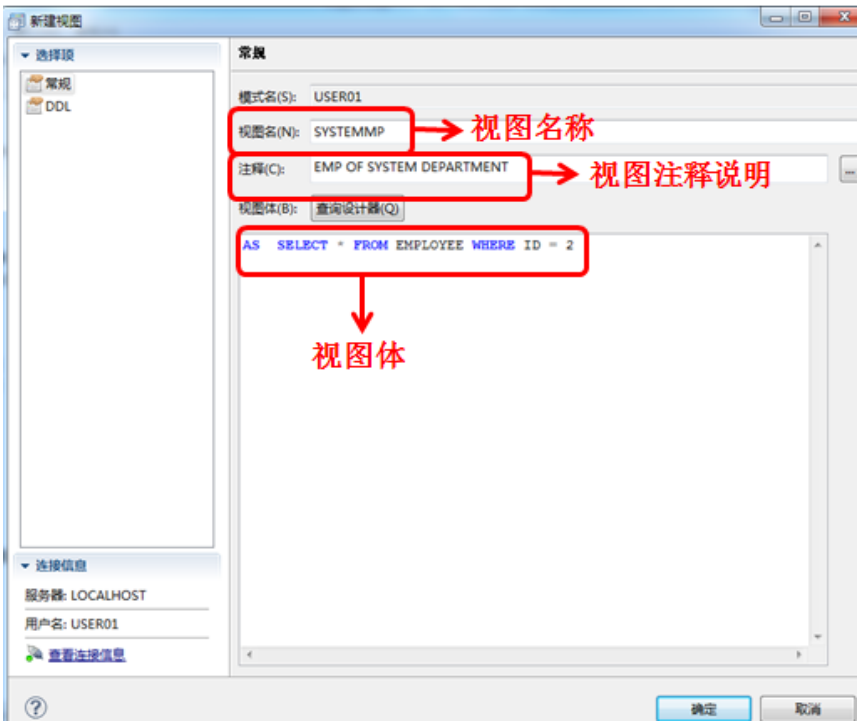
视图是从一个或几个基表（或视图）导出的表，它是一个虚表，即数据字典中只存放视图的定义，而不存放对应的数据，这些数据仍存放在原来的基表中。当需要使用视图时，则执行其对应的查询语句，所导出的结果即为视图的数据。因此当基表中的数据发生变化时，从视图中查询出的数据也随之改变了。

可以使用 DM 管理工具或者 DDL 语句 CREATE VIEW 创建视图。下面介绍用 DM 管理工具创建视图。

例：使用 DM 管理工具创建视图：

1. 展开**模式：**显示数据库中的所有模式
2. 展开 **SYSDBA：**显示该模式下的对象，包括表、视图等
3. 在**视图**上右键：显示菜单项

1、 点击**新建视图**
2、 视图名:键入 SYSTEMMP
3、 注释: 键入 EMP OF SYSTEM DEPARTMENT
4、 视图体:键入 AS　SELECT * FROM EMPLOYEE WHERE ID = 2 ;
5、 点击**确定：**创建视图 SYSTEMMP 成功
6、 操作截图



4) 序列对象

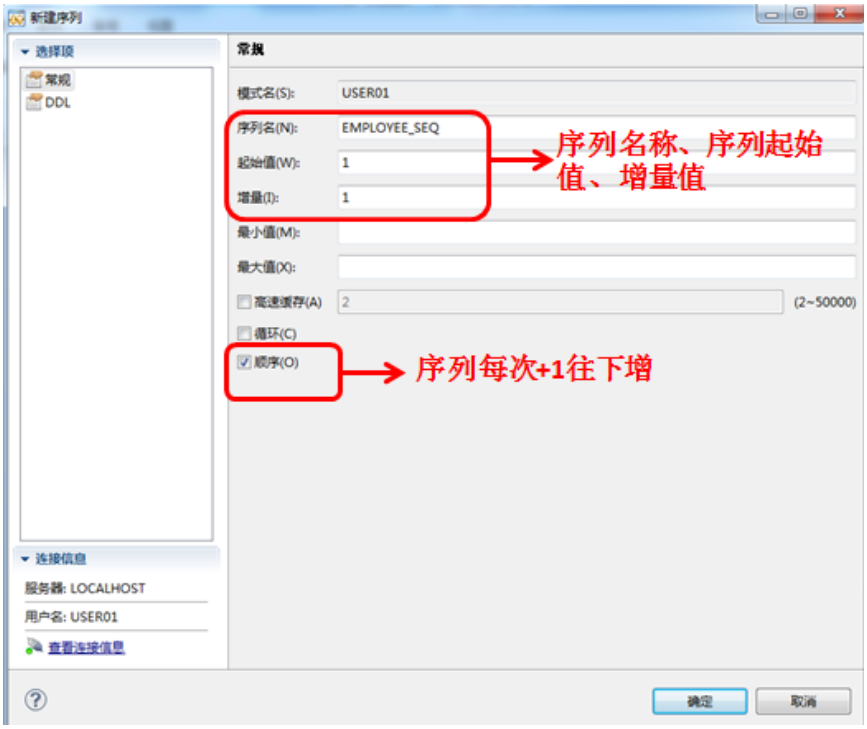
序列（SEQUENCE）是 DM 数据库中的数据库实体之一。通过使用序列，多个用户可以产生和使用一组不重复的有序整数值。比如可以用序列来自动地生成主关键字值。

可以使用 DM 管理工具或者 DDL 语句 CREATE SEQUENCE 创建序列。下面介绍用 DM 管理工具创建序列

例：使用 DM 管理工具创建序列：

1. 展开**模式：**显示数据库中的所有模式
2. 展开 **SYSDBA：**显示该模式下的对象，包括表、视图等
3. 在**序列**上右键：显示菜单项

1、 点击**新建序列**
2、 序列名：键入 EMPLOYEE_SEQ
3、 起始值：键入 1
4、 增量：键入 1
5、 顺序：选中
6、 点击**确定：**创建序列 EMPLOYEE_SEQ 成功
7、 操作截图



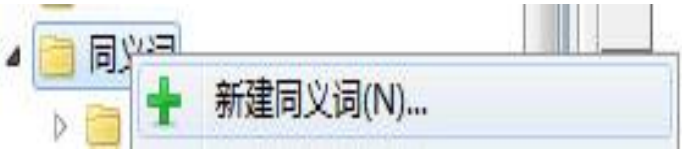
5) 同义词

同义词让用户能够为数据库的模式下的对象提供别名。同义词通过掩盖一个对象真实的名字和拥有者，并且对远程分布式的数据库对象给予了位置透明特性以此来提供了一定的安全性。同时使用同义词可以简化复杂的 SQL 语句。同义词可以替换模式下的表、视图、序列、函数、存储过程等对象。

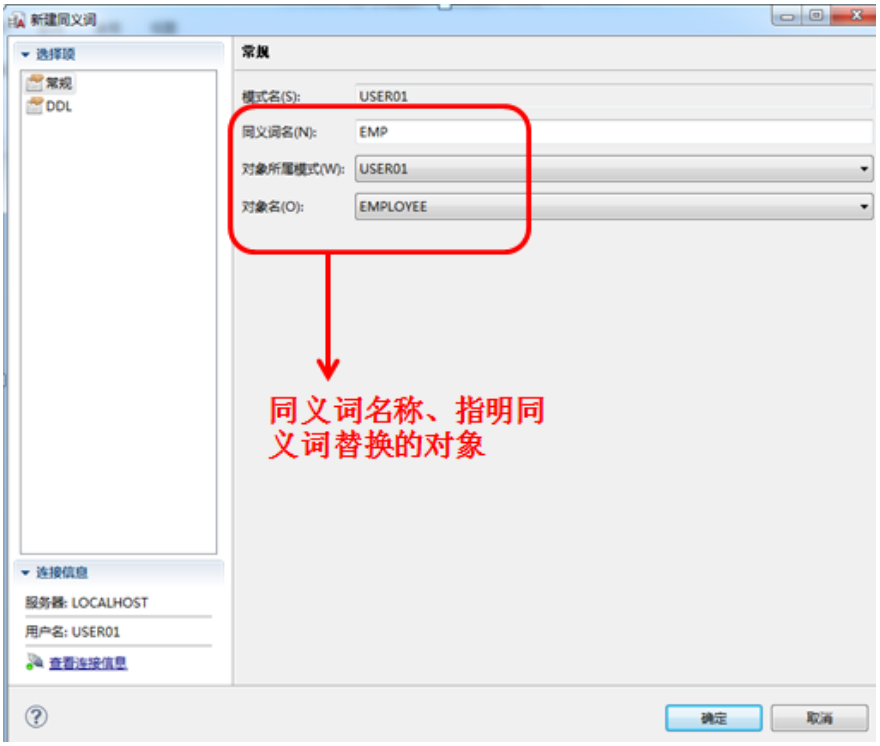
可以使用 DM 管理工具或者 DDL 语句 CREATE SYNONYM 创建序列。下面介绍用 DM 管理工具创建序列。

例：使用 DM 管理工具创建同义词：

1. 展开**模式**：显示数据库中的所有模式
2. 展开 **SYSDBA**：显示该模式下的对象，包括表、视图等
3. 在**同义词**上右键：显示菜单项



1、 点击**新建同义词**
2、 同义词名：键入 EMP
3、 对象所属模式：在下拉框中选择 USER01
4、 对象名：在下拉框中选择 EMPLOYEE
5、 点击**确定**：创建同义词 EMP 成功
6、 操作截图



6) 存储过程

用户可以通过 DM 管理工具或 CREATE PROCEDURE 语句创建存储过程。本节介绍通过管理工具 Manager 创建存储过程。

例：使用 DM 管理工具创建存储过程：

1. 展开**模式**：显示数据库中的所有模式
2. 展开 **SYSDBA**：显示该模式下的对象，包括表、视图等
3. 在**存储过程**上右键：显示菜单项

1、 点击**新建存储过程**

2、 存储过程名：键入 PROC1

3、 点击参数列表右侧的  添加按钮，对参数列表中出现的参数项进行设置，数据类型调整为 NUMBER；参数类型为 IN，不必修改；

4、 同样添加另一个参数，将数据类型修改为 VARCHAR，精度设为 50；参数类型为 IN，不必修改；

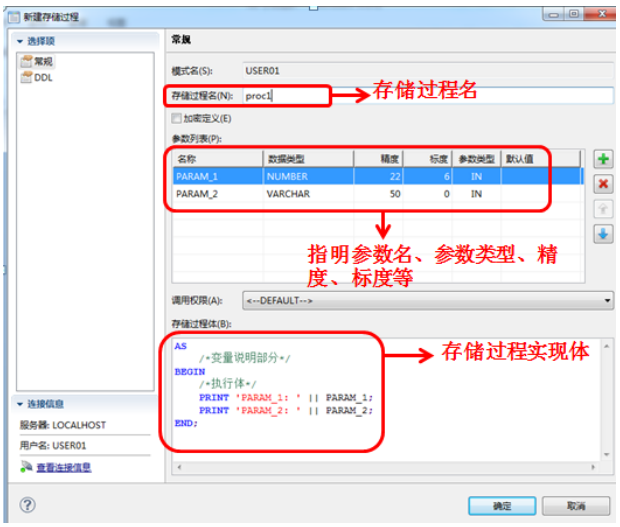
5、 使用默认的调用权限，不必修改；

6、 修改存储过程体中的文本，编写相应实现的逻辑，该测试过程将打印两个参数的值，将文本修改为：

```
AS
 /*变量说明部分*/
BEGIN
 /*执行体*/
 PRINT 'PARAM_1: ' || PARAM_1;
 PRINT 'PARAM_2: ' || PARAM_2;
END;
```

7、 点击**确定**:创建存储过程 PROC1 成功

8、 操作截图



7) 函数

用户可以通过 DM 管理工具或 CREATE FUNCTION 语句创建函数。本节介绍通过管理工具 Manager 创建函数。

例：使用 DM 管理工具创建函数

1. 展开**模式**：显示数据库中的所有模式
2. 展开 **SYSDBA**：显示该模式下的对象，包括表、视图等
3. 在**函数**上右键：显示菜单项

1、 点击**新建函数**

2、 函数名：键入 FUNC1

3、 修改参数列表中返回参数的数据类型，将其修改为 BIGINT；

4、 点击参数列表右侧的  添加按钮，对参数列表中出现的参数项进行修改，数据类型调整为 INT；参数类型为 IN，不必修改；

5、 同样添加另一个参数，将数据类型修改为 INT，参数类型为 IN，不必修改；

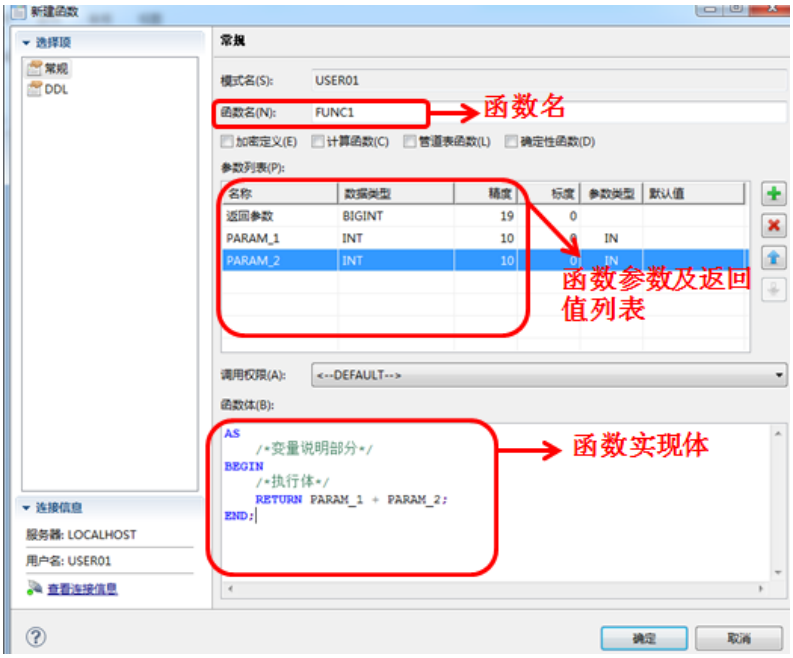
6、 使用默认的调用权限，不必修改；

7、 修改函数体中的文本，编写相应实现的逻辑，该测试函数返回两个参数的和，将文本修改为：

```
AS
 /*变量说明部分*/
 BEGIN
 /*执行体*/
 RETURN PARAM_1 + PARAM_2;
END;
```

8、 点击**确定**：创建函数 FUNC1 成功

9、 操作截图



8) 包

DM7 支持 PL/SQL 包来扩展数据库功能，用户可以通过包来创建应用程序或者使用包来管理过程和函数。本节以在 SYSDBA 模式下创建包 PKG1 为例，通过 DM 管理工具创建包的过程。用户也可以通过 CREATE PACKAGE 语句及 CREATE PACKAGE BODY 语句创建包头包体。

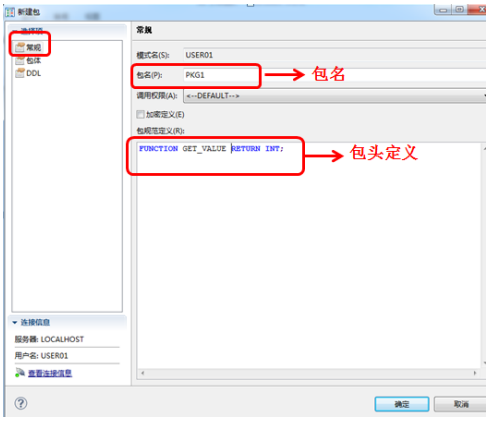
例：使用 DM 管理工具创建包

1. 展开**模式**：显示数据库中的所有模式
2. 展开 **SYSDBA**：显示该模式下的对象，包括表、视图等
3. 在**包**上右键：显示菜单项

1、 点击**新建包**

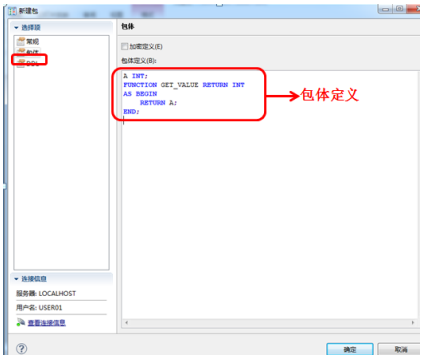
2、 选择项中选择<常规>，填写包名 PKG1

3、 选择项中选择<常规>，在<包规范定义>中填写定义的类型、变量、函数等。定义过程: FUNCTION GET_VALUE RETURN INT;



4、 点击选择项中的<包体>，在<体定义>中填写包头中定义的函数或过程的实现。在包体中声明一个包私有的变量，使用过程获取该私有变量的值。将文本修改为：

```
A INT;
FUNCTION GET_VALUE RETURN INT
AS BEGIN
    RETURN A;
END;
```



5、 点击**确定**:创建包 PKG1 成功。

9) 触发器

触发器(TRIGGER)定义为当某些与数据库有关的事件发生时，数据库应该采取的操作。这些事件包括全局对象、数据库下某个模式、模式下某个基表上的 INSERT、DELETE 和 UPDATE 操作。触发器是在相关的事件发生时由服务器自动地隐式地激发。

例: 创建一个触发器，该触发器是一个 UPDATE 前触发器，其目的是做更新操作时不论是否更新表中某列的值，该列的值保持原值不变。该测例使用 DISQL 工具进行实验操作，便于实验结果对比。

1、 使用 DM 管理工具创建表并插入数据:

```
CREATE TABLE students(sno int,grade int);
Insert into students values(100,81);
commit;
```

2、 使用 DM 管理工具创建触发器

```
CREATE OR REPLACE TRIGGER trig
BEFORE UPDATE ON students
FOR EACH ROW
BEGIN
:new.grade:=:old.grade;
END;
/
```

3、 验证触发

未UPDATE前查询 : SELECT grade FROM students WHERE sno=100;

查询结果显示: GRADE

81

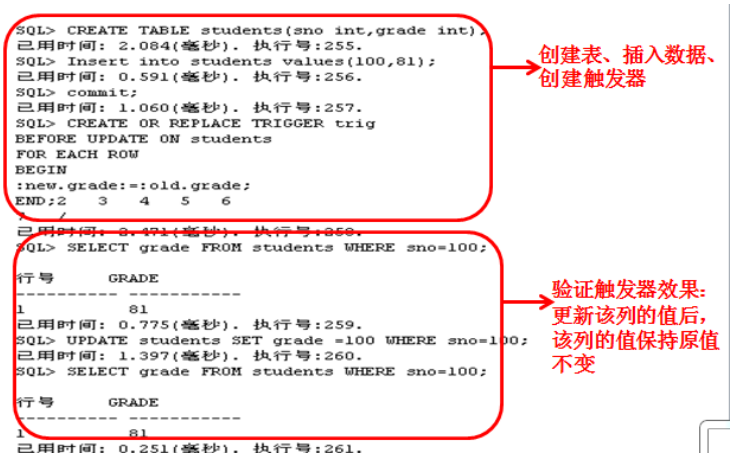
UPDATE 更新: UPDATE students SET grade =100 WHERE sno=100;

UPDATE 表以后再次查询: SELECT grade FROM students WHERE sno=100;

查询结果保持不变,查询结果：GRADE

81

4、 测试截图



更多详细的管理数据库对象相关信息，请查看 DM7_SQL 语言使用手册，或者联系在线客服。