Video 11-4：动态SQL

动态SQL

- 在编译阶段无法获得完整的SQL语句，需要在程序执行时才能够确定的SQL语句，称为“动态嵌入式SQL”
  - 允许在程序运行过程中临时“组装”出SQL语句
  - 支持动态组装SQL语句和动态参数两种形式

动态SQL（续）

1. 使用SQL语句主变量

2. 使用动态参数

1. 使用主变量

- SQL语句主变量
  - 程序主变量包含的内容是SQL语句本身的内容，而不是原来保存数据的输入或输出变量
  - SQL语句主变量在程序执行期间可以赋值为不同的SQL语句，然后执行

使用SQL语句主变量（续）

- [例8.6] 创建基本表test

```
EXEC SQL BEGIN DECLARE SECTION;
        const char *stmt="CREATE TABLE test(a int);";
      /*SQL语句主变量，内容是创建表的SQL语句*/
EXEC SQL END DECLARE SECTION;
...
EXEC SQL EXECUTE IMMEDIATE :stmt;
      /*执行动态SQL语句*/
```

2. 动态参数

- 动态参数
  - SQL语句中的可变元素
  - 使用参数符号（?）表示该位置的数据在运行时设定
- 和主变量的区别
  - 动态参数的输入不是编译时完成绑定
  - 而是通过 PREPARE语句准备主变量和执行语句 EXECUTE绑定数据或主变量来完成

动态参数（续）

- 使用动态参数的步骤

（1）声明SQL语句主变量

（2）准备SQL语句（PREPARE）

EXEC SQL PREPARE <语句名>

FROM <SQL语句主变量>;

(3) 执行SQL语句（带参数）

- EXEC SQL EXECUTE <语句名>

[INTO <主变量表>]

[USING <主变量或常量>];

例子

❖ [例8.7] 向test中插入元组。

```
EXEC SQL BEGIN DECLARE SECTION;
        const char *stmt = "INSERT INTO test VALUES(?);";
                                /*声明SQL主变量内容是INSERT语句 */
EXEC SQL END DECLARE SECTION;
...
EXEC SQL PREPARE mystmt FROM :stmt; /*准备语句*/
...
EXEC SQL EXECUTE mystmt USING 100;
                        /*执行语句，设定INSERT语句插入值100 */
EXEC SQL EXECUTE mystmt USING 200;
                        /* 执行语句，设定INSERT语句插入值200 */
```