

8.4 嵌入式SQL与存储过程

主要考点

- 1、嵌入式SQL
- 2、游标
- 3、存储过程

嵌入式SQL

- SQL提供了将SQL语句嵌入到某种高级语言中的方式，通常采用预编译的方法。

1、区分主语言与SQL语句的方式：

EXEC SQL <SQL语句>

2、向主语言传递SQL语句执行的状态信息的方式：

SQLCA，即SQL通信区，是系统默认定义的全局变量。

3、主变量（共享变量）：

- 主语言通过主变量向SQL语句提供参数，主变量是由主语言的程序定义的，并用SQL的DECLARE语句说明。
- 在SQL语句中，为了与SQL中的属性名区分，在引用共享变量时，前面需要加“：”

嵌入式SQL

例：

```
EXEC SQL BEGIN DECLARE SECTION;  
    char Mname[4], Msex[3], givensno[5];  
    int Mage;  
    char SQLSTATE[6];           //特殊的共享变量，解释SQL语句的执行状况  
EXEC SQL END DECLARE SECTION;
```

(1) 根据共享变量givensno值查询学生关系students中学生的姓名、年龄和性别。

```
EXEC SQL SELECT sname, age, sex  
    INTO :Mname, :Mage, :Msex  
    FROM students  
    WHERE sno = :givensno;
```

(2) 某学生选修了一门课程，将其插入学生选课表SC中，假设学号、课程号、成绩已分别赋给主变量HSno、Hcno 和 Hgrade。

```
EXEC SQL INSERT INTO SC(Sno, Cno, Grade)  
    Values (:HSno, :HCno, :Hgrade);
```


2009年下午试题五：

某停车场有多个入口和出口，车辆进入时从入口处由系统查询可用的停车位，从出口驶出时系统将其刚使用的车位标记为停车位。

假设实现停车场管理的伪指令如下表所示：

伪指令	说明
Get()	返回一个空车位号。如果当前没有空车位，则返回空值NULL。 例如：x=Get()，表示读取空的停车位到变量x中
Writ(A,0)	置停车位A状态为空
Writ(A,1)	置停车位A状态为非空

【问题3】（4分）

若停车位表的关系模式为：park(parkno,isused)，其中parkno为停车位号，isused为停车位标志，0为空，1为非空。

下面是用E-SQL实现的查询空车位的函数Get()，请补全空缺处的代码。

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
```

```
EXEC SQL DECLARE getblk CURSOR FOR
```

```
    (a) ;
```

```
EXEC SQL OPEN getblk ;
```

```
EXEC SQL FETCH getblk INTO :Hparkno ;
```

//Hparkno为已声明的主变量

```
IF SQLCA.sqlcode = 100 THEN
```

```
    EXEC SQL CLOSE getblk ;
```

```
    Return NULL ;
```

```
ELSE
```

```
    (b) ;
```

```
END IF
```


存储过程

- 存储过程(Procedure)：是一组为了完成特定功能的SQL语句集合，经编译后存储在数据库中，用户通过指定存储过程的名称并给出参数来执行。
- 存储过程中可以包含逻辑控制语句和数据操纵语句，它可以接受参数、输出参数、返回单个或多个结果集以及返回值。
- 由于存储过程在创建时即在数据库服务器上进行了编译并存储在数据库中，所以存储过程运行要比单个的SQL语句块要快。
- 语句格式：

```
CREATE PROCEDURE 存储过程名 (IN|OUT|IN OUT 参数1 数据类型, IN|OUT|IN OUT 参数2 数据类型...)
[AS]      //参数的数据类型只需要指明类型名即可，不需要指定宽度。具体宽度由外部调用者决定
BEGIN
    <SQL语句>
END
```

IN： 为默认值，表示该参数为输入型参数，在过程体中值一般不变。

OUT： 表示该参数为输出参数，可以作为存储过程的输出结果，供外部调用者使用。

IN OUT： 既可作为输入参数，也可作为输出参数。

2015年下午试题五：

某航空售票系统负责所有本地起飞航班的机票销售，并设有多个机票销售网点。以下为E-SQL编写的部分售票代码：

```
...  
EXEC SQL SELECT balance INTO :x FROM tickets WHERE flight = :flightno;  
    printf(" 航班%当前剩余机票数为：%d\n请输入购票数：" , flightno, x);  
    scanf("%d", &a);  
EXEC SQL UPDATE tickets SET balance = :x - :a WHERE flight = :flightno;  
...
```

【问题3】（4分）

下面是改写的存储过程，其中flightno为航班号；a为购票数；result为执行状态；1表示成功，0表示失败；表tickets中的剩余机票数balance具有大于等于零约束。请补充完整。

```
CREATE PROCEDURE buy_ticket ( char[ ] flightno IN, _____ (a) _____ , int result OUT )  
    AS  
    BEGIN  
        .....  
        UPDATE tickets SET balance = _____ (b) _____  
            WHERE flight = flightno ;  
        if (SQLcode <> SUCCESS) {                //SQLcode 为 SQL语句的执行状态  
            _____ (c) _____ ;  
            result=0; return;  
        }  
        COMMIT;  
        _____ (d) _____  
    END
```

- (a) INT a IN
- (b) balance - a
- (c) ROLLBACK WORK
- (d) result=1; return;