Video 11-3：游标Cursor

游标

- 游标
  - 游标是数据库系统为用户开设的一个数据缓冲区，存放SQL语句的执行结果
  - 每个游标区都有一个名字，也可以理解为该数据区的指针
  - 可以用SQL语句逐一从游标中（指针所指示的位置）获取记录，并赋给主变量，交由主语言进一步处理

游标

- 为什么要使用游标
  - SQL语言与主语言具有不同数据处理方式
  - SQL语言是面向集合的，一条SQL语句原则上可以产生或处理多条记录
  - 主语言是面向记录的，一组主变量一次只能存放一条记录
  - 仅使用主变量并不能完全满足SQL语句向应用程序输出数据的要求
  - 嵌入式SQL引入了游标的概念，用来协调这两种不同的处理方式

必须使用游标的SQL语句

- 必须使用游标的SQL语句
  - 查询结果为多条记录的SELECT语句
  - CURRENT形式的UPDATE语句
  - CURRENT形式的DELETE语句

1. 查询结果为多条记录的SELECT语句

- 使用游标的步骤
  - （1）申明游标
  - （2）打开游标
  - （3）推进游标指针并取当前记录
  - （4）关闭游标

（1）申明游标

- 使用DECLARE语句
- 语句格式

```
EXEC SQL DECLARE <游标名> CURSOR
        FOR <SELECT语句>;
```

- 功能
  - ■是一条说明性语句，这时关系数据库管理系统并不执行SELECT语句，而是向系统申请一个数据空间，用于存放未来执行select的结果数据集。

（2）打开游标

- 使用OPEN语句
- 语句格式

  EXEC SQL OPEN <游标名>;

- 功能
  - ■打开游标实际上是执行相应的SELECT语句，把查询结果取到缓冲区中
  - ■这时游标处于活动状态，指针指向查询结果集中的第一条记录

（3）推进游标指针并取当前记录

- 使用FETCH语句
- 语句格式

```
EXEC SQL FETCH <游标名>
        INTO <主变量>[<指示变量>]
             [,<主变量>[<指示变量>]]...;
```

- 功能
  - ■指定方向推动游标指针，同时将缓冲区中的当前记录取出来送至主变量供主语言进一步处理

（4）关闭游标

- 使用CLOSE语句
- 语句格式

  EXEC SQL CLOSE <游标名>;
- 功能
  - ■关闭游标，释放结果集占用的缓冲区及其他资源
- 说明
  - ■游标被关闭后，就不再和原来的查询结果集相联系
  - ■被关闭的游标可以再次被打开，与新的查询结果相联系

2. CURRENT形式的UPDATE/DELETE语句

- 非CURRENT形式的UPDATE语句和DELETE语句，由于是数据库一次即可完成的操作，无需与主程序的交互，因此，无需使用游标
  - 非CURRENT形式的UPDATE语句和DELETE语句
    - 面向集合的操作
    - 一次修改或删除所有满足条件的记录

CURRENT形式的UPDATE/DELETE语句（续）

- CURRENT形式的UPDATE语句和DELETE语句的用途（续）
  - 如果只想修改或删除其中某个记录
    - 用带游标的SELECT语句查出所有满足条件的记录
    - 从中进一步找出要修改或删除的记录
    - 用CURRENT形式的UPDATE语句和DELETE语句修改或删除之
    - UPDATE语句和DELETE语句中要用子句
      WHERE CURRENT OF <游标名>
      表示修改或删除的是最近一次取出的记录，即游标指针指向的记录

CURRENT形式的UPDATE语句和DELETE语句（续）

- 不能使用CURRENT形式的UPDATE语句和DELETE语句
  - 当游标定义中的SELECT语句带有UNION或ORDER BY子句
  - 该SELECT语句相当于定义了一个不可更新的视图
- 上述情形意味着无法将修改传递到数据库的基表修改上去。

3. 程序实例

❖ [例8.1] 依次检查某个系的学生记录，交互式更新某些学生年龄。

```
EXEC SQL BEGIN DECLARE SECTION;     /*主变量说明开始*/
    char Deptname[20];
    char Hsno[9];
    char Hsname[20];
    char Hssex[2];
    int HSage;
    int NEWAGE;
EXEC SQL END DECLARE SECTION;       /*主变量说明结束*/
long SQLCODE;
EXEC SQL INCLUDE SQLCA;             /*定义SQL通信区*/
```

程序实例（续）

```
int main(void)                                          /*C语言主程序开始*/
{       int  count = 0;
        char  yn;                                       /*变量yn代表yes或no*/
        printf("Please choose the department name(CS/MA/IS): ");
        scanf("%s",deptname);                   /*为主变量deptname赋值*/
        EXEC SQL CONNECT TO TEST@localhost:54321 USER
                "SYSTEM"/"MANAGER";             /*连接数据库TEST*/
        EXEC SQL DECLARE SX CURSOR FOR          /*定义游标SX*/
                SELECT Sno,Sname,Ssex,Sage      /*SX对应的语句*/
                FROM Student
                WHERE SDept = :deptname;
        EXEC SQL OPEN SX;       /*打开游标SX，指向查询结果的第一行*/
```

程序实例（续）

```
for ( ; ; )                                          /*用循环结构逐条处理结果集中的记录*/
{       EXEC SQL FETCH SX INTO :HSno,:Hsname,:HSsex,:HSage;
                                                     /*推进游标，将当前数据放入主变量*/
        if (SQLCA.SQLCODE!= 0)      /*SQLCODE != 0，表示操作不成功*/
                break;              /*利用SQLCA中的状态信息决定何时退出循环*/
        if(count++ == 0)                /*如果是第一行的话，先打出行头*/
            printf("\n%-10s %-20s %-10s %-10s\n",
                        "Sno“,"Sname“,"Ssex", "Sage");
        printf("%-10s %-20s %-10s %-10d\n“,
                    HSno,Hsname,Hssex,HSage);            /*打印查询结果*/
         printf(“UPDATE AGE(y/n)?”);    /*询问用户是否要更新该学生的年龄*/
         do{scanf("%c",&yn);}
          while(yn != 'N' && yn != 'n' && yn != 'Y' && yn != 'y');
```

.

程序实例（续）

```
        if (yn == 'y' || yn == 'Y')                    /*如果选择更新操作*/
            {
           printf("INPUT NEW AGE:");
              scanf("%d",&NEWAGE);        /*用户输入新年龄到主变量中*/
           EXEC SQL UPDATE Student          /*嵌入式SQL更新语句*/
                    SET Sage = :NEWAGE
                      WHERE CURRENT OF SX;
             }                          /*对当前游标指向的学生年龄进行更新*/
 }
  EXEC SQL CLOSE SX;          /*关闭游标SX，不再和查询结果对应*/
  EXEC SQL COMMIT WORK;                                  /*提交更新*/
  EXEC SQL DISCONNECT TEST;                    /*断开数据库连接*/
}
```