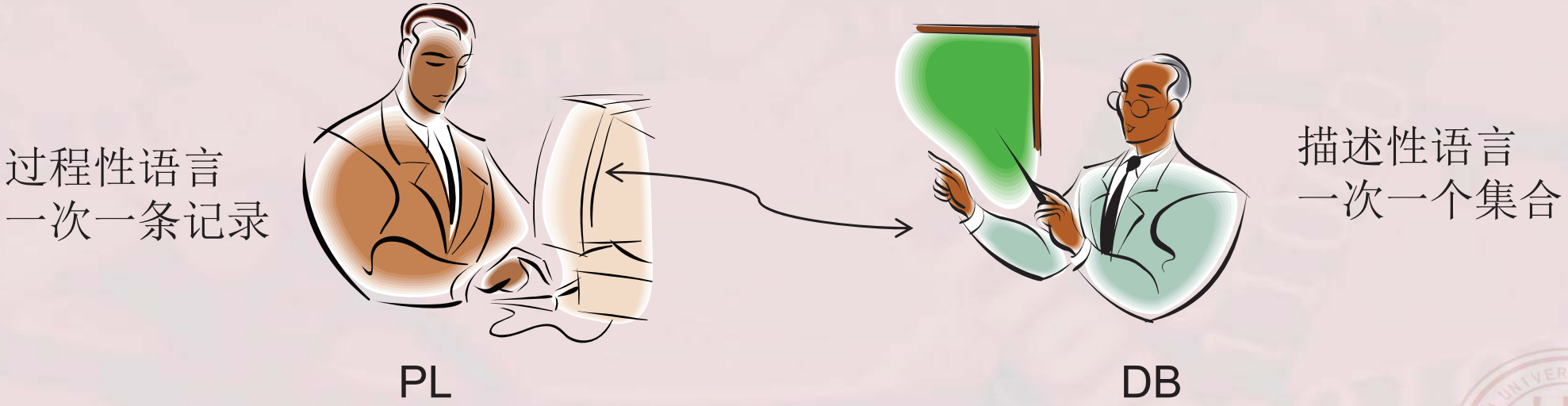
Video 11-2：通信机制

嵌入式SQL语句与主语言之间的通信

- SQL语言和高级语言是两种不同风格、不同计算模型的语言，如何对话？



嵌入式SQL语句与主语言之间的通信（续）

- 数据库工作单元与主程序工作单元之间的通信

（1）向主语言传递SQL语句的执行状态信息，使主语言能够据此控制程序流程，主要用SQL通信区实现

（2）主语言向SQL语句提供参数，主要用主变量实现

（3）将SQL语句查询数据库的结果返回给主语言处理，主要用主变量和游标实现

嵌入式SQL语句与主语言之间的通信

1. SQL通信区

- SQLCA的用途
  - SQL语句执行后，系统反馈给应用程序信息
    - 描述系统当前工作状态
    - 描述运行环境
  - 这些信息将送到SQL通信区中
  - 应用程序从SQL通信区中取出这些状态信息，据此决定接下来执行的语句

1. SQL通信区（2）

- SQLCA： SQL Communication Area
  - SQLCA是一个数据结构

```
struct sqlca_t
{
  char    sqlcaid[8];
  long    sqlabc;
  long    sqlcode;
  struct
  {
    int      sqlerrml;
    char     sqlerrmc[SQLERRMC_LEN];
  }     sqlerrm;
  char    sqlerrp[8];
  long    sqlerrd[6];
  char    sqlwarn[8];
  char    sqlstate[5];
};
```

1. SQL通信区（3）

- SQLCA使用方法
  - 定义SQLCA
    - 用EXEC SQL INCLUDE SQLCA定义
  - 使用SQLCA
    - SQLCA中有一个存放每次执行SQL语句后返回代码的变量SQLCODE
    - 如果SQLCODE等于预定义的常量SUCCESS，则表示SQL语句成功，否则表示出错
    - 应用程序每执行完一条SQL 语句之后都应该测试一下SQLCODE的值，以了解该SQL语句执行情况并做相应处理

2. 主变量

- 主变量
  - 嵌入式SQL语句中可以使用主语言的程序变量来输入或输出数据
  - 在SQL语句中使用的主语言程序变量简称为主变量（Host Variable）

主变量（续）

- 主变量的类型
  - 输入主变量
    - 由应用程序对其赋值，SQL语句引用
  - 输出主变量
    - 由SQL语句对其赋值或设置状态信息，返回给应用程序

主变量（续）

- 指示变量
  - 是一个整型变量，用来“指示”所指主变量的值或条件
  - 一个主变量可以附带一个指示变量（Indicator Variable）
  - 指示变量的用途
    - 指示输入主变量是否为空值
    - 检测输出变量是否为空值，值是否被截断

SELECT语句（续）

- [例8.3] 查询某个学生选修某门课程的成绩。假设已经把将要查询的学生的学号赋给了主变量givensno，将课程号赋给了主变量givencno。

```
EXEC SQL SELECT Sno,Cno,Grade
        INTO :Hsno,:Hcno,:Hgrade:Gradeid  /*指示变量Gradeid*/
        FROM  SC
        WHERE Sno=:givensno AND Cno=:givencno;
```

如果Gradeid < 0，不论Hgrade为何值，均认为该学生成绩为空值。

增删改语句（续）

- [例8.5] 某个学生新选修了某门课程，将有关记录插入SC表中。假设插入的学号已赋给主变量stdno，课程号已赋给主变量couno。

```
gradeid=-1；                          /*gradeid为指示变量，赋为负值*/
EXEC SQL INSERT
        INTO SC(Sno,Cno,Grade)
        VALUES(:stdno,:couno,:gr :gradeid)；
                                      /*:stdno，:couno，:gr为主变量*/
```

由于该学生刚选修课程，成绩应为空，所以要把指示变量赋为负值

主变量（续）

- 在SQL语句中使用主变量和指示变量的方法
  - ■说明主变量和指示变量

```
BEGIN DECLARE SECTION
        ...
        ...                    （说明主变量和指示变量）
        ...
END DECLARE SECTION
```

主变量（续）

- 在SQL语句中使用主变量和指示变量的方法（续）
  - 使用主变量
    - 说明之后的主变量可以在SQL语句中任何一个能够使用表达式的地方出现
    - 为了与数据库对象名（表名、视图名、列名等）区别，SQL语句中的主变量名前要加冒号（:）作为标志
  - 使用指示变量
    - 指示变量前也必须加冒号标志
    - 必须紧跟在所指主变量之后

主变量（续）

- 在SQL语句之外（主语言语句中）使用主变量和指示变量的方法
  - 直接使用，不必加冒号
  - 这一点必须特别注意

3 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

3 游标

❖ 使用DECLARE语句可申明一个游标

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。

3. 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

4. 例子

```
        EXEC SQL BEGIN DECLARE SECTION;    /*主变量说明开始*/
            char deptname[20];
        EXEC SQL END DECLARE SECTION;      /*主变量说明结束*/
        long SQLCODE;
        EXEC SQL INCLUDE SQLCA;            /*定义SQL通信区*/
int main(void)                                    /*C语言主程序开始*/
{           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*/
```

思考题：比较以下概念

1 主语言的变量

2 主变量

3 SQL通信区中的变量