

数据库系统概论

An Introduction to Database System

第八章 数据库编程

中国人民大学信息学院

SQL的局限

尽管**SQL**语言是关系数据库的标准化语言，功能强大。但是，仍然存在许多的要求无法表达。举例：

- 1 查询一个班级里位于中位的学生成绩
- 2 计算一个班级学生的学分绩



突破SQL局限性

突破SQL语言局限性的可行的技术方案:

- 1 利用高级语言的表达能力: 嵌入式SQL
- 2 扩展SQL语言对于过程控制的表达能力: 过程化SQL
- 3 在一个更大的视野上, 将数据库看做是一类数据源: ODBC编程



第11讲：嵌入式SQL

（本讲 共有4个视频，对应教科书8.1节）



Video 11-1: 初识嵌入式SQL



初识嵌入式SQL

```
#include <stdio.h>
#include <stdlib.h>
```

```
EXEC SQL BEGIN DECLARE SECTION; /**主变量说明开始 */
```

```
char HSno[9];
char HSname[20];
char HSex[2];
int HSage;
char Hdept[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() /**c语言主程序开始 */
```

```
{
```

```
printf("Please input the sno:");/**输入要查找的学生学号 */
scanf("%s", HSno);
```

```
/** 连接数据库（数据库名为TEST，主机名为localhost，
端口号为54321，用户名密码为“SYSTEM/krms” */
```

```
EXEC SQL CONNECT TO TEST@localhost:54321 AS CONN1
USER "SYSTEM" USING "manager";
```

```
EXEC SQL SELECT Sno,Sname,Ssex,Sage,Sdept
INTO:HSno,:Hname,:Hsex,:Hage,:Hdept
FROM Student
WHERE Sno=:HSno;
```

```
if (sqlca.sqlcode == 0)
```

```
{
```

```
printf("\n% - 9s% - 20s% - 2s% - 4s% - 20s\n", "Sno",
"Sname", "Ssex", "Sage", "Sdept");
printf("% - 9s% - 20s% - 2s% - 4s% - 20s\n", HSno,
Hname, Hsex, HSage, Hdept);
}
```

```
EXEC SQL DISCONNECT CONN1;
```

```
return 0;
```

```
}
```

嵌入式SQL语句的基本格式

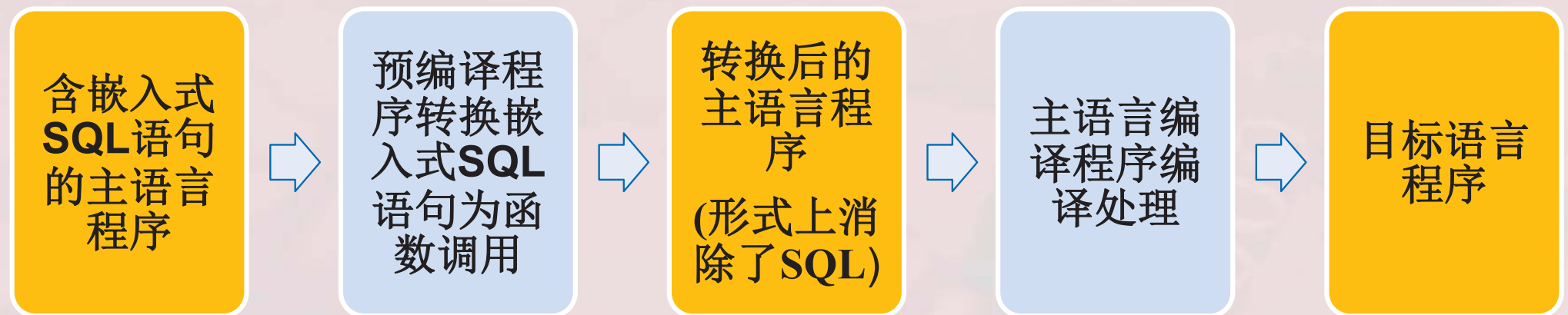
❖ 为了区分SQL语句与主语言语句，所有SQL语句必须加前缀**EXEC SQL**，

主语言为C语言时，语句格式：

■ **EXEC SQL <SQL语句>;**

❖ 举例：**EXEC SQL SELECT Sno,Sname,Ssex,Sage,Sdept
INTO:Hsno,:Hname,:Hsex,:Hage,:Hdept
FROM Student
WHERE Sno=:givensno;**

嵌入式SQL的处理过程（续）



从主语言访问数据库的基本步骤

- ❖ 第一步：打开数据库；
- ❖ 第二步：定义必要的主变量和数据通信区；
- ❖ 第三步：用**SQL**访问数据库，并对返回结果进行处理；
- ❖ 第四步：关闭数据库



1 建立数据库连接

**EXEC SQL CONNECT TO target [AS connection-name]
[USER user-name];**

- **target**是要连接的数据库服务器的名字
 - 常见的服务器标识串，如<dbname>@<hostname>:<port>
 - 包含服务器标识的**SQL**串常量
 - **DEFAULT**



建立数据库连接（续）

**EXEC SQL CONNECT TO target [AS connection-name]
[USER user-name];**

■ **connect-name**是可选的连接名

- 连接名必须是一个有效的标识符
- 在整个程序内只有一个连接时可以不指定连接名
- 程序运行过程中可以修改当前连接

**EXEC SQL SET CONNECTION connection-name
|DEFAULT;**



2 定义主变量与数据通讯区

```
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*/
```



3 执行SQL语句

❖ 所有的**SQL**语句都可以以嵌入式方式使用：

- 数据定义语句 (**CREATE**)
- 数据控制语句 (**GRANT,REVOKE, COMMIT, ROBACK**)
- 数据更新语句 (**UPDATE, INSERT, DELETE**)
- 数据查询语句 (**SELECT**)



SELECT语句

❖ [例8.2] 根据学生号码查询学生信息。

```
EXEC SQL SELECT Sno,Sname,Ssex,Sage,Sdept  
          INTO:Hsno,:Hname,:Hsex,:Hage,:Hdept  
          FROM Student  
          WHERE Sno=:givensno;
```

/* 增加了一个INTO子句，用于保存查询结果 */

/*前面带“:”标识的变量称为主变量，也就是主语言的变量 */



SELECT语句（续）

- ❖ INTO子句、WHERE子句和HAVING短语的条件表达式中均可以使用主变量
- ❖ 查询返回的记录中，可能某些列为空值NULL
- ❖ 注意：如果查询结果实际上并不是单条记录，而是多条记录，**则程序出错**，关系数据库管理系统会返回错误信息（见视频11-3）



增删改语句

❖ 在UPDATE的SET子句和WHERE子句中可以使用主变量

❖ [例8.4] 修改某个学生选修1号课程的成绩。

```
EXEC SQL UPDATE SC
```

```
    SET Grade=:newgrade
```

```
        /*修改的成绩已赋给主变量: newgrade*/
```

```
WHERE Sno=:givensno AND Cno= “1” ;
```

```
        /*学号赋给主变量: givensno*/
```



4 关闭数据库连接

EXEC SQL DISCONNECT [connection];



5. 程序实例

❖ [例8.1] 打印指定学号的学生记录

```
EXEC SQL BEGIN DECLARE SECTION;  
    char Deptname[20];  
    char HSNo[9];  
    char HSName[20];  
    char HSSex[2];  
    int HSAge;  
EXEC SQL END DECLARE SECTION;
```

主变量定义区

```
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*/*

```
    printf("Please input the student number ");
```

```
    scanf("%s", Hsno);
```

*/*为主变量Hsno赋值*/*

```
EXEC SQL CONNECT TO TEST@localhost:54321 AS CONN1
USER "SYSTEM" USING "MANAGER";
```

连接Test数据库

```
EXEC SQL SELECT Sno,Sname,Ssex,Sage,Sdept
INTO:Hsno,:Hname,:Hsex,:Hage,:Hdept
FROM Student
WHERE Sno=:HSno;
```

执行SQL查询



程序实例（续）

```
if (sqlca.sqlcode == 0) /* SQLCA中的SQLCODE == 0 表示操作成功 */  
{  
    printf("\n% - 9s% - 20s% - 2s% - 4s% - 20s\n", "Sno", "Sname", "Ssex",  
        "Sage", "Sdept");  
    printf("% - 9s% - 20s% - 2s% - 4s% - 20s\n", Hsno, Hname, Hsex, HSage,  
        Hdept);  
}
```

```
EXEC SQL DISCONNECT CONN1;}
```

关闭游标，断开连接

