



# 数据库编程概述

## 总述

内容：在DBMS和数据库应用系统中都可以使用编程的方法对数据库中的数据进行操纵。本知识点学习数据库编程的基本概念。

- 一、SQL语言的运行环境
- 二、在应用程序中使用SQL

## 一、SQL语言的运行环境

SQL语言有交互式 and 嵌入式两种不同使用方式。

### (1) 交互式使用

在终端上直接交互使用SQL，称为交互式SQL。

SQLQuery1.sql - PA...ministrator

```
SELECT *  
FROM S;
```

结果 消息

	sno	sname	age	sex
1	S1	王红	17	男
2	S2	李明	19	女
3	S3	王小非	21	女
4	S4	张静	20	男

你好，管理员

### 新书信息

[新书入库](#)  
[办理借书证](#)  
[图书借阅](#)  
[图书归还](#)  
[查看图书或读者信息](#)  
[修改读者信息](#)  
[修改图书信息](#)  
[图书丢失](#)  
[借书证挂失](#)  
[逾期未还图书](#)

书名:  \*

作者

出版社

类别编号

价格

入库数量

## 一、SQL语言的运行环境

### (2) 嵌入式使用

数据库应用系统开发的主要方法是嵌入式SQL。将SQL语句嵌入到其它宿主语言编写的程序中，作为宿主语言的子语言，使宿主语言具备访问数据库的能力。

## 二、在应用程序中使用SQL

### (1) 语句级接口

SQL结构在宿主语言中以新的语句类型出现，首先由各DBMS厂商提供的预编译器，将SQL结构转换成宿主语言能够处理的过程调用，再用宿主语言编译器编译。一般将该种方式称为嵌入式SQL。

## 二、在应用程序中使用SQL

### (1) 语句级接口

- 语法简单易学
- 执行效率较高

宿主语言 + 嵌入式SQL

预处理程序

宿主语言 + 函数调用

宿主语言编译程序

宿主语言目标程序

## 二、在应用程序中使用SQL

### (2) 调用级接口

通过一个公共、与DB无关的应用程序设计接口。

- 由于不同DBMS厂商使用的SQL语句和通信协议都略有不同，使得在某个DBMS下编写的应用程序不能在另一个DBMS下运行。
- 越来越多的应用程序需要共享多个部门的数据资源，访问不同的DBMS。



## 二、在应用程序中使用SQL

### (2) 调用级接口

应用程序完全以宿主语言编写，SQL语句是在运行时生成的字符串变量的值，作为参数传递给宿主语言，不需要预编译。

- ODBC (开放数据库互连)
- JDBC (JAVA数据库互连)

## 练习

以下（ ）方法不是应用程序实现数据库编程的方法。

- A. 嵌入式SQL
- B. ODBC
- C. JDBC
- D. 交互式SQL

## 解答

以下（ D ）方法不是应用程序实现数据库编程的方法。

- A. 嵌入式SQL
- B. ODBC
- C. JDBC
- D. 交互式SQL

## 小结

SQL既可以交互式执行，也可以嵌入在程序中使用。前台开发语言和后台数据库共同完成数据库应用系统的开发工作。高级语言负责实现屏幕控制、窗口和菜单管理以及报表生成等功能，而SQL语言负责实现对后台数据库的访问。



谢谢！



# 嵌入式SQL

## 总述

内容：在DBMS和数据库应用系统中都可以使用编程的方法对数据库中的数据进行操纵。本知识点学习使用嵌入式SQL进行数据库编程的基本方法。

- 一、嵌入式SQL定义
- 二、主语言中SQL语句的识别
- 三、主语言与SQL语句之间的通信

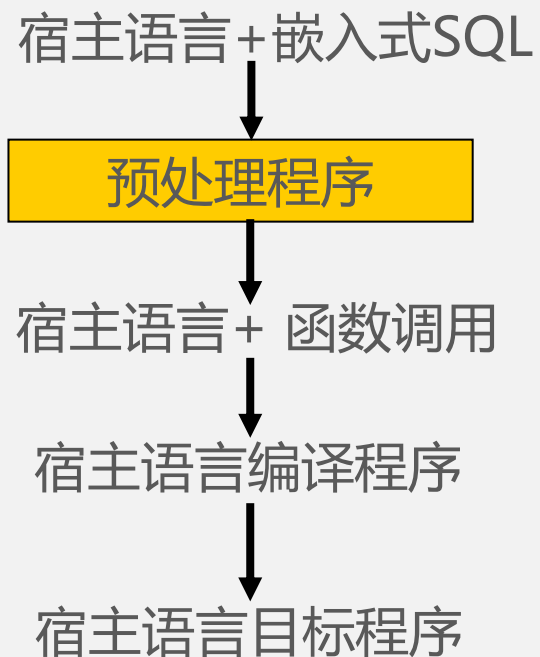


## 一、嵌入式SQL定义

SQL结构在宿主语言中以新的语句类型出现，首先由各DBMS厂商提供的预编译器，将SQL结构转换成宿主语言能够处理的过程调用，再用宿主语言编译器编译。一般将该种方式称为嵌入式SQL。

## 一、嵌入式SQL定义

嵌入式SQL基本处理过程



## 一、嵌入式SQL定义

嵌入式SQL的两个问题

- (1) 如何识别主语言中的SQL语句？
- (2) SQL语句负责操纵数据库，而高级语言负责控制程序流程。它们之间如何通信？

## 二、主语言中SQL语句的识别

将主语言中的SQL语句置于EXEC SQL 和  
END SQL ( ; ) 之间。

```
main()
```

```
{ EXEC SQL SELECT :sname=sname FROM S  
      WHERE sno=: sno ;  
  printf( "学生姓名: " ,sname) ;  
}
```

## 三、主语言与SQL语句之间的通信

- (1) 用主变量（共享变量）实现：主语言向SQL语句提供参数。将SQL语句中使用的主语言程序变量称为主变量。

```
main()  
{ EXEC SQL SELECT sname INTO :sname  
      FROM S WHERE sno = :sno;  
  printf( "学生姓名: " ,sname);  
}
```

## 三、主语言与SQL语句之间的通信

- (1) 用主变量（共享变量）实现：主语言向SQL语句提供参数。将SQL语句中使用的主语言程序变量称为主变量。

```
BEGIN DECLARE SECTION  
    STRING kk1, kk2  
END DECLARE SECTION
```

} 声明

```
EXEC SQL SELECT sname INTO :kk1 FROM S  
WHERE sno=:kk2;
```

## 三、主语言与SQL语句之间的通信

(2) 用主变量和游标实现：将SQL语句查询数据库的结果交给主语言进一步处理。

由于SQL语句处理的是记录集合，而主语句一次只能处理一个记录，因此需要用游标来协调这两种不同的处理方式，把集合操作转换成单记录处理方式。

## 三、主语言与SQL语句之间的通信

(3) 用SQL通信区 (SQLCA) 实现：向主语言传递SQL语句的执行状态信息，使主语言能够据此信息控制程序流程。

SQL语句执行后，系统将当前的工作状态和运行环境的各种数据送到SQL通信区 (SQLCA) 中，应用程序从SQLCA中取出状态信息，决定接下来执行的语句。



## 三、主语言与SQL语句之间的通信

SQLCA中有一个变量`SQLCODE`，存放每次执行SQL语句后返回的代码。应用程序执行完一条SQL语句后，测试SQLCODE的值了解SQL语句是否成功执行，以决定下一步如何处理，控制程序流程。

如果SQL语句成功，则SQLCODE等于0；如果SQL语句失败，则SQLCODE不等于0，存放错误代码。

不同DBMS的SQLCA结构定义不同，也可以使用SQLSTATE存放错误代码。

## 三、主语言与SQL语句之间的通信

例1：在S表中查找某个学生

...

EXEC SQL INCLUDE SQLCA;           (1) 定义SQL通信区

EXEC SQL BEGIN DECLARE SECTION;   (2) 主变量说明

    CHAR sname(10);

EXEC SQL END DECLARE SECTION;

main( )

{ EXEC SQL SELECT sname INTO :sname FROM S  
    WHERE sno = 's1' ;

    if (SQLCA.sqlcode<>0) printf( "没有该学生" )

        else printf( "学生姓名: " ,sname) }

## 练习

嵌入式SQL中，使用（     ）实现向主语言传递SQL执行状态信息。

- A. 主变量
- B. 游标
- C. SQLCA
- D. 指示变量

## 解答

嵌入式SQL中，使用（ C ）实现向主语言传递SQL执行状态信息。

- A. 主变量
- B. 游标
- C. SQLCA
- D. 指示变量

## 小结

在嵌入式SQL中，SQL语句与主语言语句分工明确。SQL语句直接与数据库打交道，取出数据库中的数据。主语言语句控制程序流程，对取出的数据做进一步加工处理。

嵌入式SQL的实现方法是，用特定标识来识别SQL语句，用主变量、游标和SQLCA通信区进行信息交换。



谢谢！

The background of the slide features a light blue, semi-transparent overlay with a grid pattern. A large, white, pixelated cursor icon is positioned in the lower right corner. The overall aesthetic is clean and modern, typical of a presentation slide.

# 游标

## 总述

内容：嵌入式SQL是将SQL语句嵌入到程序设计语言中进行数据库编程的方法，在嵌入式SQL中，SQL语句与主语言之间的通信用主变量和游标实现。本知识点学习游标的概念和用法。



- 一、游标的概念
- 二、游标的使用
- 三、游标的示例

## 一、游标的概念

在嵌入式SQL中，由于SQL语句处理的是记录集合，一条SQL语句原则上可以产生或处理多条记录，而主语句一次只能处理一个记录，因此需要用游标（CURSOR）来协调这两种不同的处理方式，把集合操作转换成单记录处理方式。

## 一、游标的概念

游标是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，每个游标区有一个名字。通过游标逐一获取记录，并赋给主变量，交给主语言处理。

## 二、游标的用法

- (1) 游标定义语句 (DECLARE)
- (2) 游标打开语句 (OPEN)
- (3) 游标推进语句 (FETCH)
- (4) 游标关闭语句 (CLOSE)

## 二、游标的用法

(1) 游标定义语句 (DECLARE)

语法:

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

## 二、游标的用法

### (2) 游标打开语句 (OPEN)

语法:

**OPEN** <游标名>;

该语句执行游标定义中的SELECT语句，游标是一个指针，此时指向查询结果的第一行之前。

## 二、游标的用法

### (3) 游标推进语句 (FETCH)

语法：

**FETCH** <游标名> **INTO** <变量表> ;

此时按行推进游标，并把游标指向的行（称为当前行）中的值取出，送到主变量。

变量表是由逗号分开的主变量组成。 FETCH语句常置于宿主语言程序的循环结构中，并借助宿主语言的处理语句逐一处理查询结果中的一个元组。

## 二、游标的用法

### (4) 游标关闭语句 (CLOSE)

语法:

**CLOSE** <游标名>;

关闭游标，释放结果集占用的缓冲区和其他资源，使它不再和查询结果相联系。



## 三、游标的示例

例1：显示表SC (sno, cno, grade) 中的数据。

...

EXEC SQL INCLUDE SQLCA; (1) 定义SQL通信区

EXEC SQL BEGIN DECLARE SECTION; (2) 主变量说明

CHAR sno(4);

CHAR cno(4);

INT grade;

EXEC SQL END DECLARE SECTION;

## 三、游标的示例

Main( )

{ EXEC SQL DECLARE C1 CURSOR FOR (3) 定义游标

SELECT sno, cno, grade FROM SC;

EXEC SQL OPEN C1; (4) 打开游标

For( ; ; )

{ EXEC SQL FETCH C1 into: sno, :cno, :grade; (5) 推进游标

If ( SQLCA.sqlcode<>0 ) break;

(6) 利用SQLCA中状态信息

Printf( "%s, %s, %d" , sno, cno, grade); }

EXEC SQL CLOSE C1; (7) 关闭游标

}

## 练习

嵌入式SQL语句与主语言之间通信的主要实现方式不包括（ ）。

- A. 建立聚簇索引
- B. 主变量
- C. 游标
- D. SQL通信区

## 解答

嵌入式SQL语句与主语言之间通信的主要实现方式不包括（ A ）。

A. 建立聚簇索引

B. 主变量

C. 游标

D. SQL通信区

## 小结

SQL语言与主语言具有不同的数据处理方式，SQL是面向集合的，主语言是面向记录的，所以嵌入式SQL用游标来协调这两种不同的处理方式。

如果是说明性语句、数据定义语句、数据控制语句以及查询结果是单记录的查询语句，不需要使用游标。当查询结果是多条记录时则需要使用游标。



谢谢！



# ODBC/JDBC编程

## 总述

内容：在DBMS和数据库应用系统中都可以使用编程的方法对数据库中的数据进行操纵。本知识点学习使用ODBC/JDBC进行数据库编程的基本方法。



一、ODBC编程

二、JDBC编程

## 一、ODBC编程

由于不同的数据库管理系统的存在，在某个RDBMS下编写的应用程序就不能在另一个RDBMS下运行。

此外，许多应用程序需要共享多个部门的数据资源，访问不同的RDBMS。

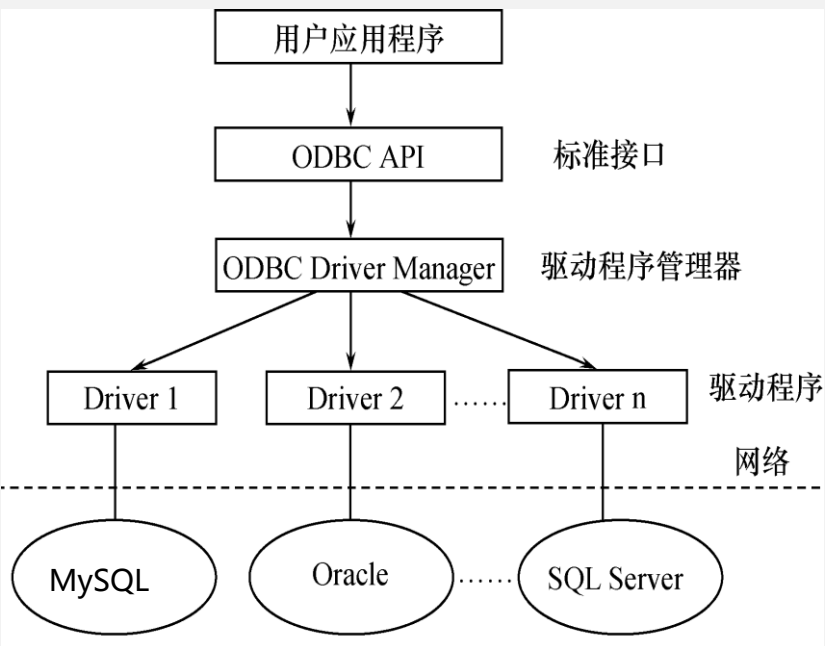
ODBC技术随之产生。

## 一、ODBC编程

ODBC (Open Database Connectivity, 开放数据库互连) 是微软公司开放服务体系 (Windows Open Services Architecture, WOSA) 中有关数据库的一个组成部分, 提供了一组访问数据库的标准API。

## 一、ODBC编程

### ODBC应用系统的体系结构



- 用户应用程序
- 驱动程序管理器
- 数据库驱动程序
- ODBC数据源管理

## 一、ODBC编程

ODBC应用程序包括的内容:

- 请求连接数据库;
- 向数据源发送SQL语句;
- 获取数据库操作结果, 或处理错误;
- 进行数据处理并向用户提交处理结果;
- 断开与数据源的连接。

## 一、ODBC编程

驱动程序管理器：

- 装载ODBC驱动程序；
- 选择和连接正确的驱动程序；
- 管理数据源；
- 检查ODBC调用参数合法性等。

当一个应用程序请求对其数据源的连接，驱动程序管理器读取该数据源的描述，定位并加载适当的驱动程序，管理应用程序和驱动程序的连接和通信。

## 一、ODBC编程

驱动程序：

- 各种操作请求由驱动程序管理器提交给某个RDBMS的ODBC驱动程序；
- 通过调用驱动程序所支持的函数来存取数据库；
- 数据库的操作结果也通过驱动程序返回给应用程序；
- 如果应用程序要操纵不同的数据库，就要动态地链接到不同的驱动程序上。

ODBC应用程序不能直接存取数据库，ODBC通过驱动程序来提供应用系统与数据库平台的独立性。

## 一、ODBC编程

数据源：

- ODBC给每个被访问的数据源指定唯一的数据源名 (Data Source Name, 简称DSN) ;
- 在连接中, 用数据源名来代表用户名、服务器名、所连接的数据库名等;
- 最终用户无需知道DBMS或其他数据管理软件、网络以及有关ODBC驱动程序细节。

数据源是最终用户需要访问的数据, 包含了数据库位置和数据库类型等信息, 是一种数据连接的抽象, 数据源对最终用户是透明的。



## 一、ODBC编程

开发ODBC应用程序的典型步骤

- (1) 为程序所需的语句、连接、环境声明句柄变量 (SQLHSTMT、SQLHDBC、SQLHENV类型) 和返回变量 (SQLRETURN类型) ;
- (2) 在程序中建立一个环境记录和一个连接记录 (使用函数SQLAllocHandle) ;
- (3) 建立一个到特定数据库服务器的连接 (使用函数SQLConnect) ;
- (4) 在程序中建立一个语句记录 (使用函数SQLAllocHandle) ;

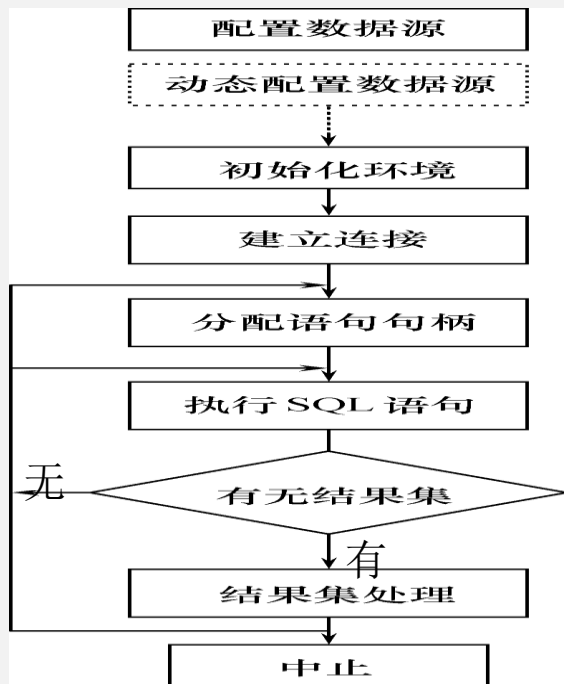
## 一、ODBC编程

### 开发ODBC应用程序的典型步骤

- (5) **准备语句**（使用函数SQLPrepare），**将各个输入参数和程序变量绑定**（使用SQLBindParameter）；
- (6) **执行SQL语句**（使用SQLExecute函数）；
- (7) **将查询结果的各个列与程序变量绑定**（使用SQLBindCol函数）；
- (8) **将列值逐一送给程序变量，交给程序处理**（使用SQLFetch函数）；
- (9) 最后，依次**释放语句句柄、连接句柄和环境句柄**。

## 一、ODBC编程

### ODBC的工作流程



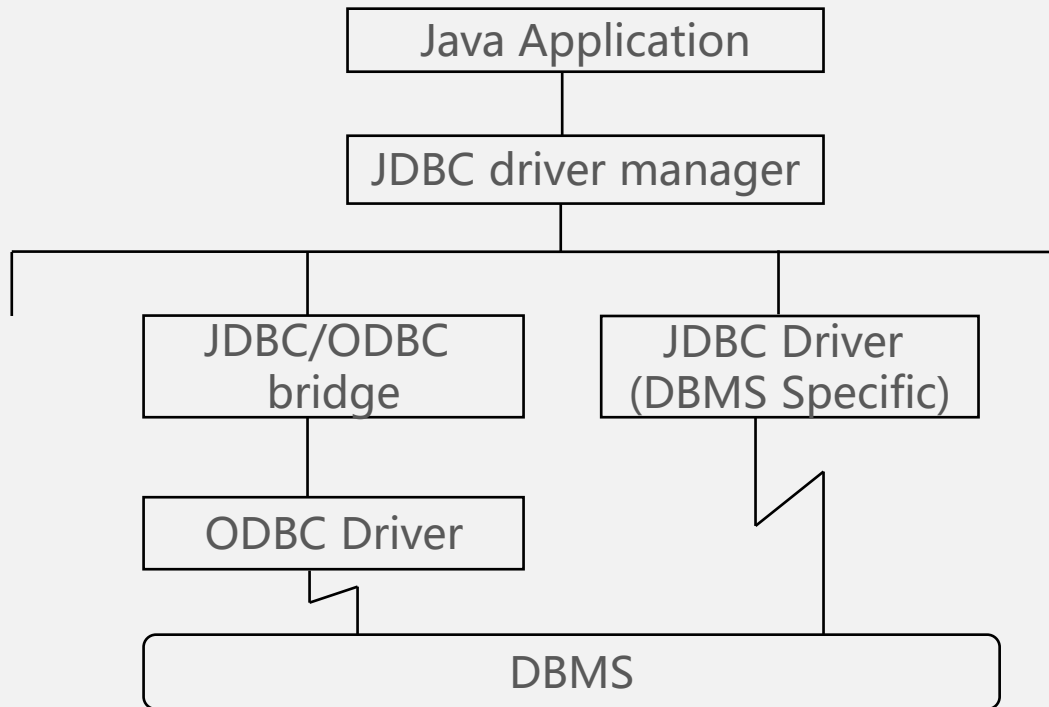
## 二、JDBC编程

JDBC (Java DataBase Connectivity, Java数据库连接) 由sun公司开发, 是JAVA语言的一部分, 提供在JAVA程序中执行SQL语句的调用级接口 (访问数据库的一组API) 。

JDBC编程方法与ODBC类似。

## 二、JDBC编程

### JDBC应用系统的体系结构



## 二、JDBC编程

### JDBC Code示例

```
try { Class.forName ( "oracle.jdbc.driver.OracleDriver" );  
    //加载Oracle提供的JDBC驱动程序  
    Connection conn = DriverManager.getConnection(  
        "jdbc:oracle:thin:@aura.bell-labs.com:2000:bankdb",  
        userid, passwd);    //连接数据源  
    Statement stmt = conn.createStatement();  
    //创建 Statements 类对象,用来发送简单SQL语句  
    ... Do Actual Work ....  
    stmt.close(); //关闭Statement对象  
    conn.close(); //关闭Connection对象  
}
```

## 练习

下列任务中（ ）不是ODBC应用程序做的工作。

- A. 请求连接数据库
- B. 向数据源发送SQL语句
- C. 选择和连接正确的驱动程序
- D. 断开与数据源的连接

## 解答

下列任务中（ C ）不是ODBC应用程序做的工作。

- A. 请求连接数据库
- B. 向数据源发送SQL语句
- C. 选择和连接正确的驱动程序
- D. 断开与数据源的连接



## 小结

嵌入式SQL语法简单易学，执行效率高，但移植性较差，共享性也较差。

使用ODBC和JDBC编程，可以提高应用系统与数据库平台的独立性，使得应用系统的移植变得容易，能同时访问不同的数据库，共享多个数据资源。



谢谢！



# 存储过程

## 总述

内容：在DBMS和数据库应用系统中都可以使用编程的方法对数据库中的数据进行操纵。本知识点学习使用存储过程进行数据库编程的方法。

- 一、数据库系统的两层架构
- 二、存储过程的定义
- 三、存储过程的建立和执行
- 四、存储过程的作用

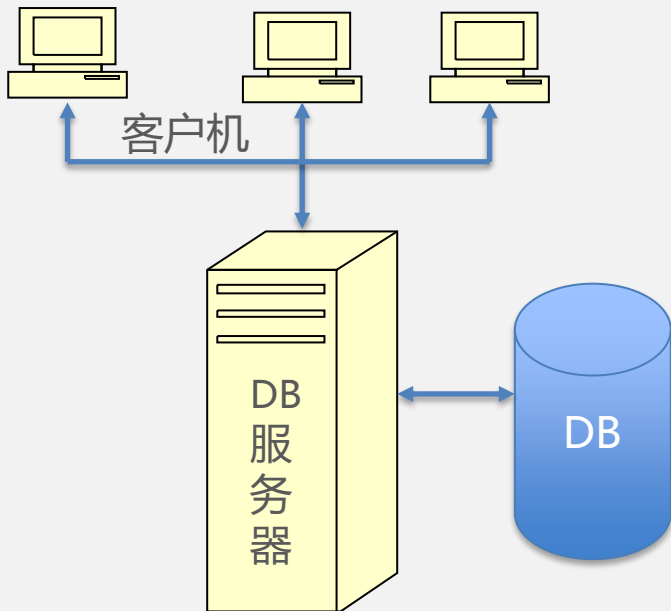
## 一、数据库系统的两层架构

从用户角度，数据库系统的结构可分为单用户结构、主从式结构、分布式结构、客户-服务器（C/S）两层结构、浏览器-应用服务器-数据库服务器（B/S）三层结构等。

在客户-服务器（C/S）结构中，客户端负责用户界面和业务逻辑，服务器端负责访问数据库。

## 一、数据库系统的两层架构

客户-服务器（C/S）结构示意图：



## 一、数据库系统的两层架构

数据库应用系统开发的主要方法

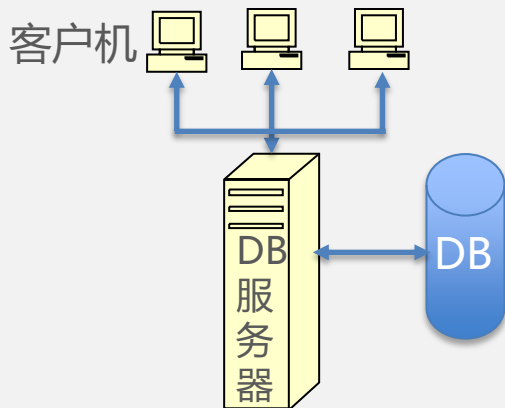
将SQL语句嵌入到其它宿主语言编写的程序中，作为宿主语言的子语言，使宿主语言具备访问数据库的能力。

部分DBMS厂商还提供了一种允许用户使用存储过程进行数据库编程的方法。



## 二、存储过程的定义

存储过程是一组SQL语句及控制流语句集合，即将常用的访问数据库的程序作为一个过程，经编译和优化后，存储在数据库服务器上，供用户调用。



## 二、存储过程的定义

存储过程是一组SQL语句及控制流语句集合，即将常用的访问数据库的程序作为一个过程，经编译和优化后，存储在数据库服务器上，供用户调用。

在 SQL Server 中建立存储过程使用的语言称为T-SQL。T-SQL由数据定义语言（DDL）、数据操纵语言（DML）、查询语言（QL）、数据控制语言（DCL）和T-SQL增加的语言元素组成。

## 二、存储过程的定义

T-SQL增加的语言元素包括：

- 常量、变量、运算符与表达式；
- 流程控制语句，包括IF ELSE、GOTO、WHILE、CONTINUE、BREAK、RETURN、WAITFOR等；
- 函数。

## 二、存储过程的定义

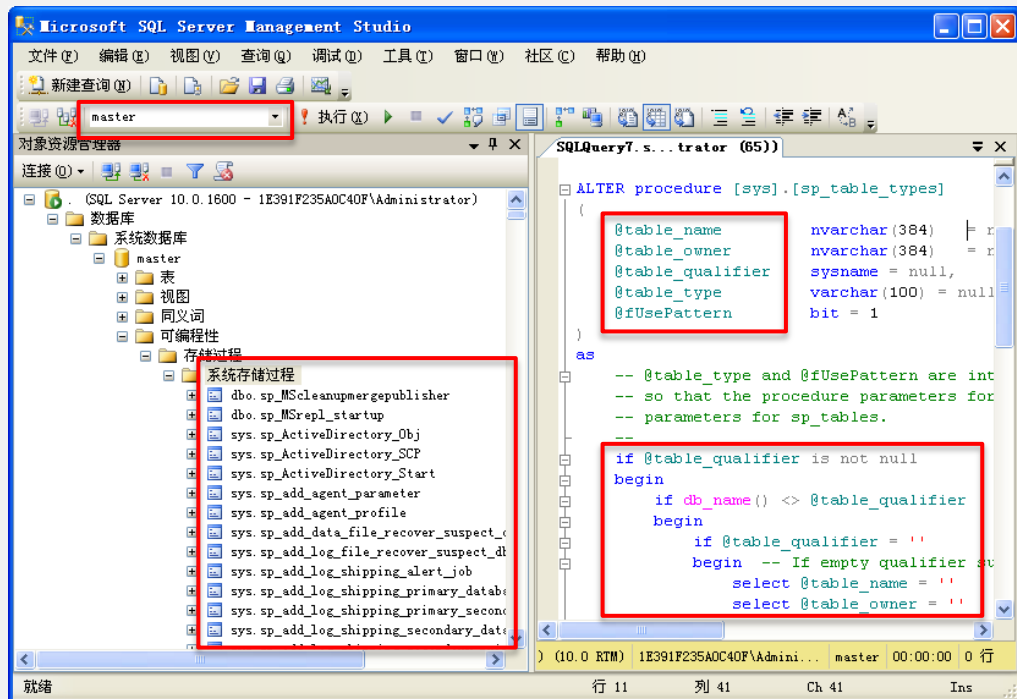
SQL Server中的存储过程

### (1) 系统存储过程

系统存储过程可作为命令执行各种操作。存放于系统数据库master中，以sp\_开头。可以从任何数据库中执行系统存储过程。

### (2) 用户自定义存储过程

## 二、存储过程的定义



## 三、存储过程的建立和执行

### (1) 建立存储过程

```
CREATE PROCEDURE <过程名>[参数...]
```

```
AS
```

```
<SQL语句组>
```

### (2) 执行存储过程

```
CALL/EXEC PROCEDURE <过程名>
```

## 三、存储过程的建立和执行

例1：编写存储过程查找某个学生某门课程的成绩。

```
CREATE PROCEDURE s_info @sname char(8),  
@cname char(10)  
AS  
    SELECT sname, cname, grade  
    FROM S, C, SC  
    WHERE S.sno=SC.sno AND C.cno=SC.cno AND  
           S.sname=@sname AND C.cname=@cname
```

执行存储过程：

```
EXEC s_info '王红' , '操作系统'
```

## 三、存储过程的建立和执行

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'student' database is expanded, showing the 'dbo.s\_info' stored procedure. The main pane shows the SQL script for creating and executing the procedure. The script is as follows:

```
SQLQuery1.sql - PA...ministrator (53))*
CREATE PROCEDURE s_info
    @sname char(8), @cname char(10)
AS
SELECT sname, cname, grade
FROM S, C, SC
WHERE S.sno=SC.sno and C.cno=SC.cno
    and S.sname=@sname and C.cname=@cname

SQLQuery1.sql - PA...ministrator (53))*
CREATE PROCEDURE s_info
    @sname char(8), @cname char(10)
AS
SELECT sname, cname, grade
FROM S, C, SC
WHERE S.sno=SC.sno and C.cno=SC.cno
    and S.sname=@sname and C.cname=@cname

exec s_info '王红', '操作系统'
```

The 'Results' tab at the bottom shows the output of the execution:

	sname	cname	grade
1	王红	操作系统	95



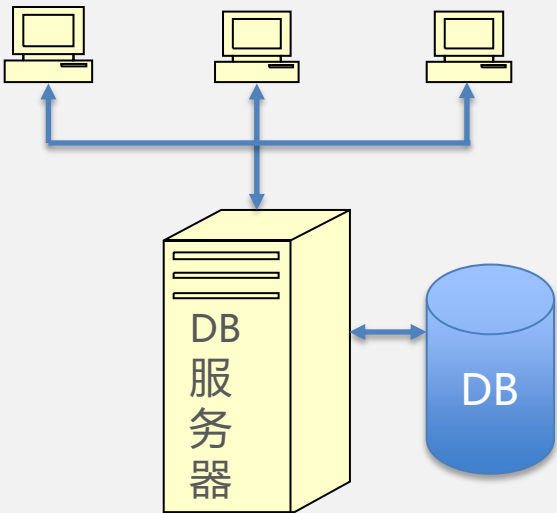
## 四、存储过程的作用

存储过程的优点：

(1) 执行速度很快（在服务器端执行，效率高）。

(2) 存储过程模块化编程，可以重复使用，减少数据库开发人员的工作量，便于维护。

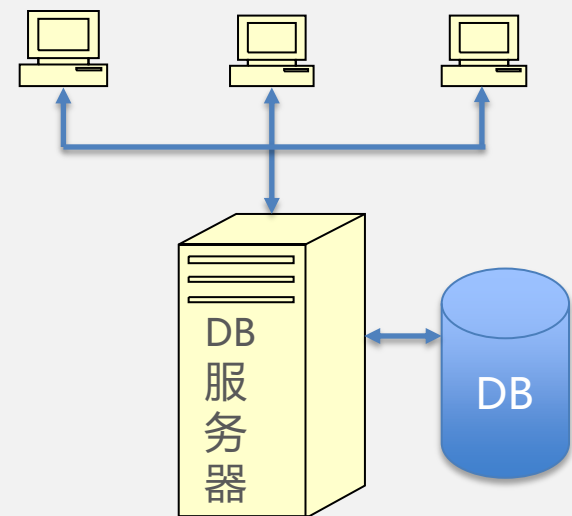
(3) 可设定只有某些用户才具有对指定存储过程的使用权，提高数据库的安全性。



## 四、存储过程的作用

存储过程的问题：

- (1) 增加了DB服务器的负载。
- (2) 可移植性差。
- (3) 一些SQL语句不能使用，调试过程不如应用程序方便。



## 练习

说明存储过程和视图的区别。

## 解答

存储过程和视图都可以存储查询，两者不同如下表。

	视图	存储过程
语句	只能是SELECT语句	可以包含程序流、逻辑以及SELECT语句
输入、返回结果	不能接受参数，只能返回结果集	可以有输入输出参数，也可以有返回值
典型应用	多个表格的连接查询	完成某个特定的较复杂的任务

## 小结

存储过程是SQL语句及控制流语句集合，在它们第一次执行时被编译，存储在数据库服务器端。

是否使用存储过程要根据具体情况而定，在环境和业务背景稳定不变的情况下，可以使用存储过程完成单个的任务，达到执行速度快、安全性好的目的。



谢谢！



# 函数

## 总述

内容：在DBMS和数据库应用系统中都可以使用编程的方法对数据库中的数据进行操纵。本知识点学习函数的使用方法。



- 一、函数的基本概念
- 二、函数的建立
- 三、函数的调用

## 一、函数的基本概念

大部分DBMS提供函数功能，方便用户的运算与操作。例如在SQL Server中主要有两类函数：

- (1) 系统内置函数
- (2) 用户自定义函数

## 一、函数的基本概念

### 系统内置函数

#### (1) 数学函数和字符串处理函数

如ABS（绝对值）、LEFT（返回左边指定个数的字符）、LTRIM（删除前导空格）等。

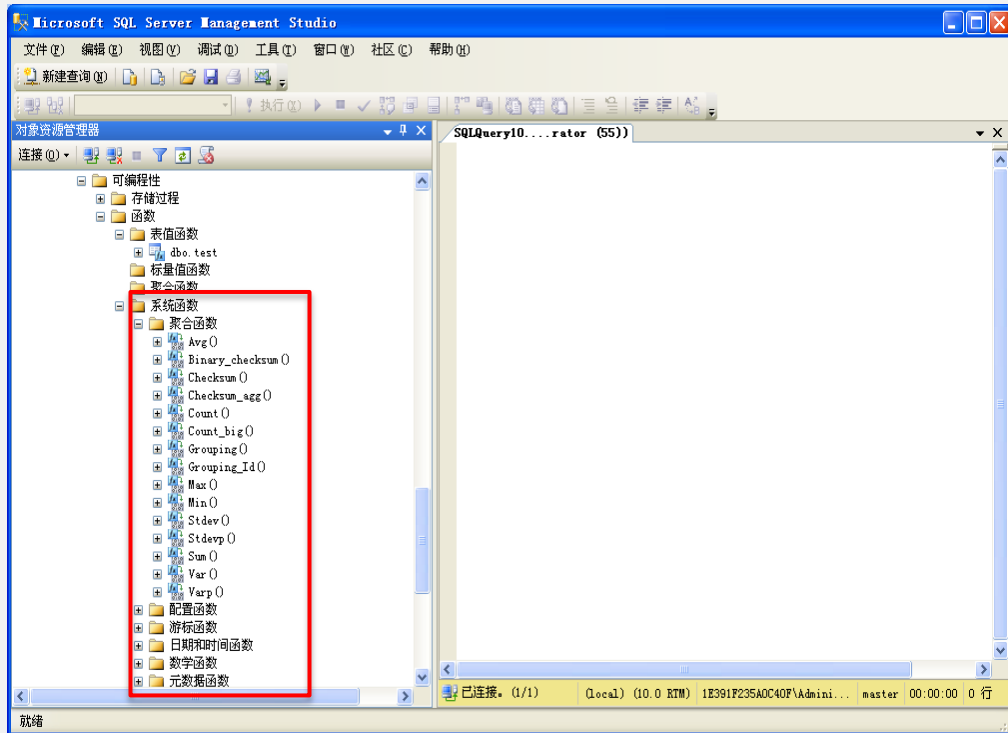
#### (2) 系统函数

如CASE（对每个WHEN求值）、CAST（数据类型转换）等。

#### (3) 日期时间函数

如GETDATE（当前日期）等。

## 一、函数的基本概念



## 二、函数的建立

### (1) 建立函数

CREATE FUNCTION 函数名( @参数名 )

RETURNS 标量值或表

AS

.....

## 二、函数的建立

例1：编写函数计算某个学生的平均成绩（标量）。

```
CREATE FUNCTION Test (@a CHAR(8)) RETURNS INT
AS BEGIN
    DECLARE @b INT
    SELECT @b=
        ( SELECT AVG(g) FROM S, SC
          WHERE S.sno=SC.sno and S.sname=@a
          GROUP BY S.sname )
    RETURN @b
END
```

## 二、函数的建立

例2：编写函数输出某个学生的各科成绩（表）。

```
CREATE FUNCTION Test (@a CHAR(4))  
    RETURNS TABLE  
AS RETURN  
    (SELECT sname,cname,g  
     FROM S,C,SC  
     WHERE S.sno=SC.sno AND  
           C.cno=SC.cno AND S.sno=@a  
    )
```

## 三、函数的调用

### (1) 标量函数的调用

SELECT 所有者名.函数名(实参)

EXEC 所有者名.函数名 实参

### (2) 表函数的调用

SELECT \* FROM 函数名 (实参)



## 三、函数的调用

例1：编写函数计算某个学生的平均成绩（标量）。

```
CREATE FUNCTION Test (@a CHAR(8)) RETURNS INT
AS BEGIN
    DECLARE @b INT
    SELECT @b=
        ( SELECT AVG(g) FROM S, SC
          WHERE S.sno=SC.sno and S.sname=@a
            GROUP BY S.sname )
    RETURN @b
END
```

```
执行:  DECLARE @b1 int
        EXEC      @b1=dbo.Test  '王一'
        SELECT    @b1
```

## 三、函数的调用

例2：编写函数输出某个学生的各科成绩（表）。

```
CREATE FUNCTION Test (@a CHAR(4))  
    RETURNS TABLE  
AS RETURN  
    (SELECT sname,cname,g  
     FROM S,C,SC  
     WHERE S.sno=SC.sno AND  
           C.cno=SC.cno AND S.sno=@a  
    )
```

执行：SELECT \* FROM Test ( 's001' )

## 练习

关于DBMS中使用的函数，不正确的说法是（ ）。

- A. 一些DBMS允许用户自定义函数
- B. 函数必须有返回值
- C. 可以在SQL语句中使用函数
- D. 函数返回值只能是数值

## 解答

关于DBMS中函数，不正确的说法是（ D ）。

- A. 一些DBMS允许用户自定义函数
- B. 函数必须有返回值
- C. 可以在SQL语句中使用函数
- D. 函数返回值只能是数值

## 小结

在存储过程、触发器、SQL查询中都可以使用函数。函数和存储过程一样，一次编译多次访问。使用函数可以使查询语言表达力强又简单，掌握常用的函数可以方便用户编程。



谢谢！