第8章　SQL Server数据库应用

8.1 SQL Server 2005 系统概述

8.2 数据库创建与程序设计

8.3 数据库应用系统设计例

8.1 SQL Server 2005 系统概述

- 1. SQL Server 2005的组成

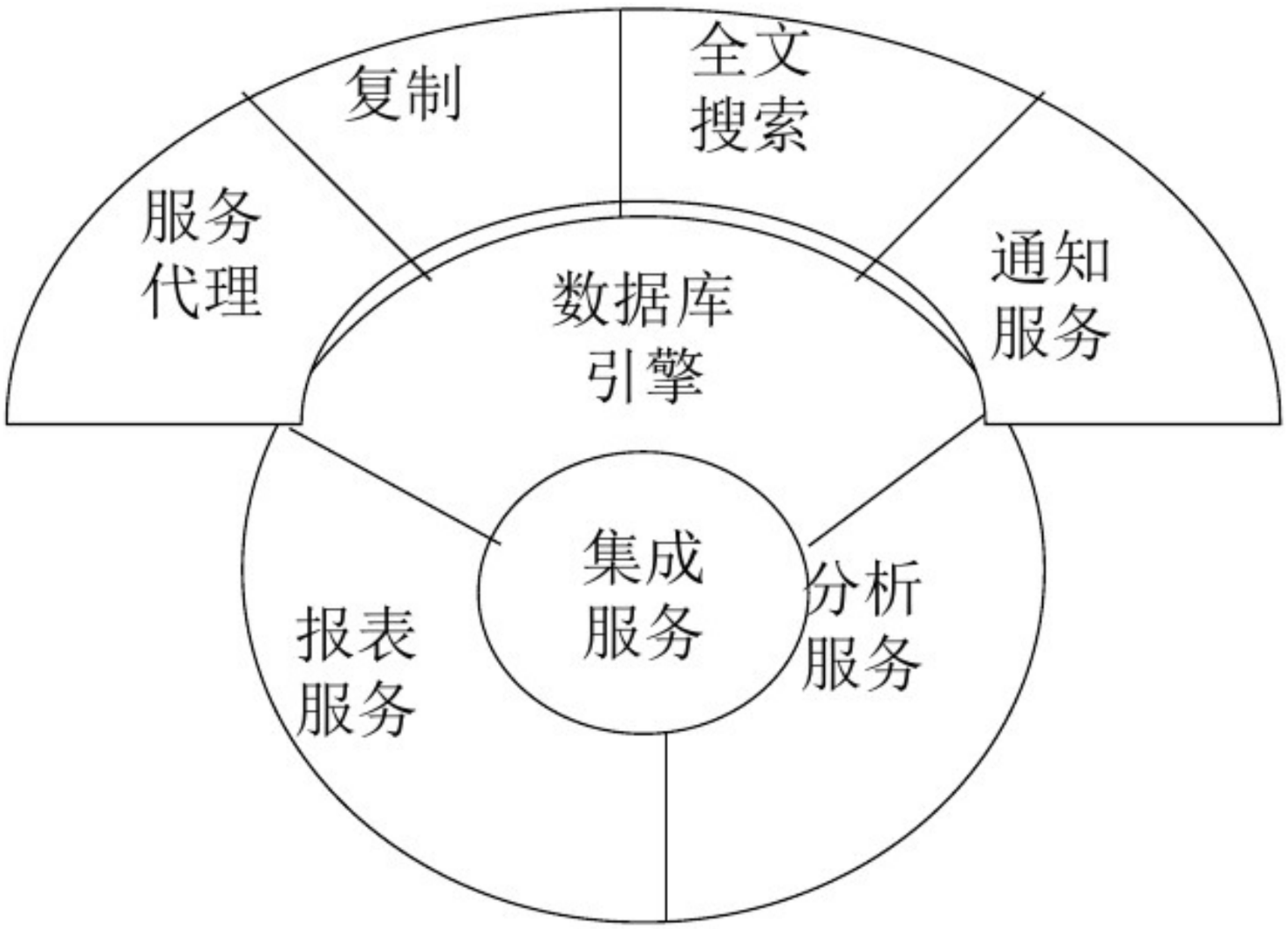


图8-1 SQL Server 2005的组成

8.1 SQL Server 2005 系统概述

- 2. SQL Server 2005的组件

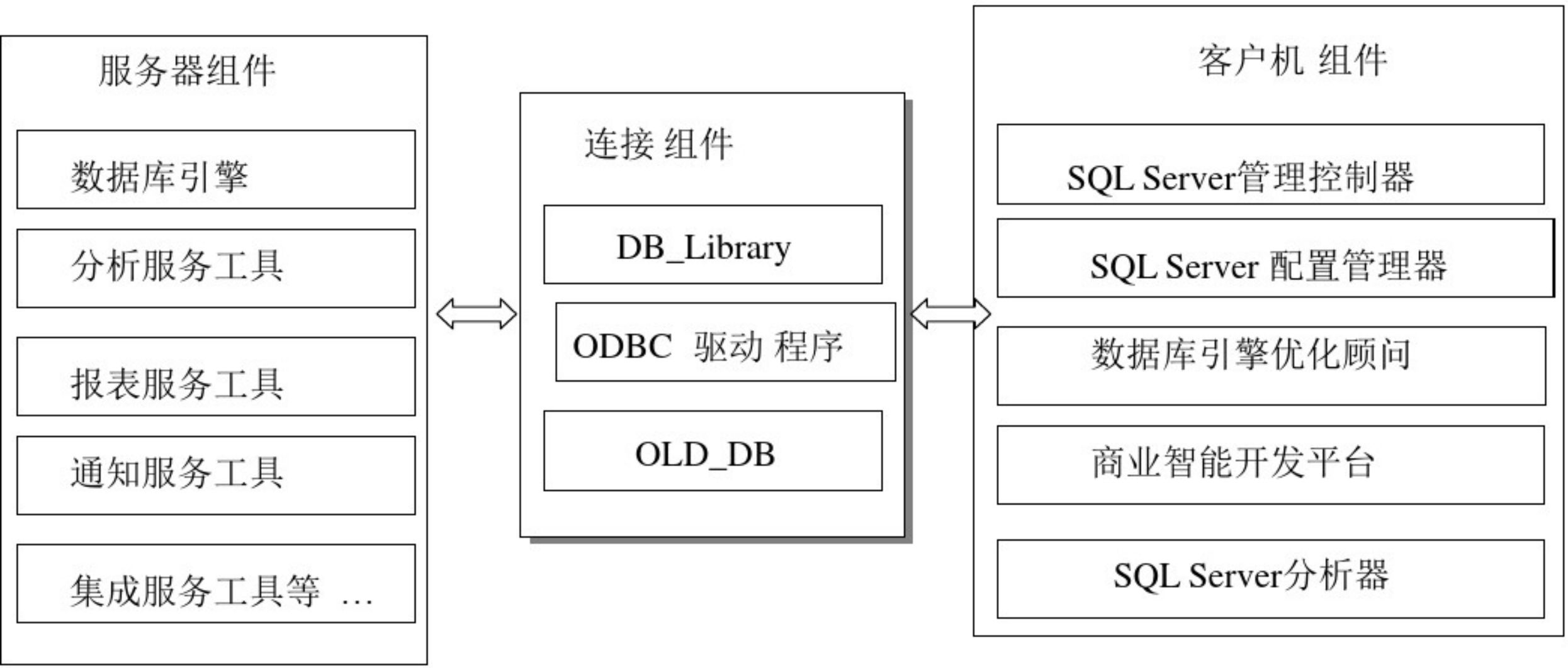


图8-2 SQL Server系统组件分布图

8.1 SQL Server 2005 系统概述

- 3 .SQL Server 2005的工具

8.2 数据库创建与程序设计

- 1. 数据库的创建
- 常使用SQL Server管理控制器建立数据库。【例8.1】
- 2 表的创建
- （1）student表。该表用于存放所有学生记录
- student（sno，sname，ssex，sbirthday，sclass）
- （2）teacher表。该表用于存放所有教师记录
- teacher（tno，tname，tsex，tbirthday，prof，depart）
- （3）course表。该表用于存放所有课程记录
- course（cno，cname，tno）
- （4）score表。该表用于存放所有学生成绩记录
- score（sno，cno，grade）
- 两种创建数据库表的方法，一种是利用SQL Server管理控制器；另一种是利用T-SQL语句中的CREATE TABLE命令。
- 【例8.2】

8.2 数据库创建与程序设计

- 3. T-SQL程序基础
- 1. 常规标识符与数据类型
- 在SQL Server中，标识符就是指用来定义服务器、数据库、数据库对象和变量等的名称。可以分为常规标识符和分隔标识符。T-SQL中不区分大小写字母。
- 数据类型包括系统数据类型和用户自定义数据类型。
- 2. 变量
- 在SQL Server中，变量分为局部变量和全局变量。
- 全局变量记录了SQL Server的各种状态信息。全局变量的名称前面为“@@”，由系统定义和维护。
- 局部变量前面有一个at符号（@），由用户定义和使用，一般格式如下：
- DECLARE {@变量名 数据类型 }
- 在SQL Server中，一次可以定义多个变量。
- 例如：DECLARE @f float,@cn char(8)

8.2 数据库创建与程序设计

- 3. 控制流语句
- T-SQL提供称为控制流的特殊关键字，用于控制T-SQL语句、语句块和存储过程的执行流。这些关键字可用于T-SQL语句、批处理和存储过程中。
- 4. 执行T-SQL语句
- 在SQL Server管理控制器中，用户可在全文窗口中输入T-SQL语句，执行语句并在结果窗口中查看结果。可以使用SQL Server管理控制器交互式地执行T-SQL语句。

控制流语句	说明
BEGIN...END	定义语句块
IF...ELSE	条件处理语句
CASE	分支语句
WHILE	循环语句
GOTO	无条件跳转语句
WAITFOR	延迟语句
BREAK	跳出循环语句
CONTINUE	重新开始循环语句

8.2 数据库创建与程序设计

- **4. 存储过程:** 是在数据库服务器端执行的一组T-SQL语句的集合，经编译后存放在数据库服务器端。
- 存储过程作为一个单元进行处理并以一个名称来标识。编程中只需要给出存储过程的名称和必需的参数，就可以方便地调用它们。
- **存储过程的分类** 【例8.7】【例8.8】
- SQL Server 2005提供了多种存储过程，分为:
- (1)用户存储过程。用户编定的可以重复使用的T-SQL语句功能模块，并且在数据库中有唯一的名称，可以附带参数。后面主要介绍用户存储过程。
- (2)系统存储过程。通常使用“sp_”为前缀，主要用于管理SQL Server和显示有关数据库及用户的信息。系统存储过程在master数据库中创建并保存，可从任何数据库中执行这些存储过程。
- (3)扩展存储过程。允许用户使用编程语言创建自己的外部例程。

【例8.7】【例8.8】 **8.3 DB应用系统设计例—见教材**

第9章 Oracle数据库应用

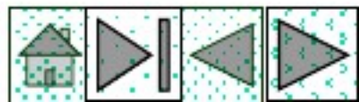
9.1 Oracle产品及功能介绍

9.2 PL/SQL程序设计

9.3 数据库应用系统设计

9.1 Oracle产品及功能介绍

- 1. 主要产品:Oracle电子商务套件；Oracle10g。
- 2. 数据库的基本概念
- **模式**是与每个Oracle数据库用户相关的一组数据库对象的集合。
- **表**是数据库中基本存储的逻辑单位，由多行和多属性（列）组成。
- **视图**可从表(或其他视图)中派生出来。
- **包**是存储过程和函数的封装，由一组相关的存储过程和函数组成。
- **触发器**是完成特定功能的程序单元。
- **数据块**是Oracle服务器所能分配、读取或写入的最小存储单位。
- **区**是DB存储空间分配的逻辑单位，由许多连续的数据块组成。
- **段**是构成表空间的逻辑存储结构，段由一组区组成。
- **表空间**是数据库中最大的逻辑单位。
- **序列生成器**可产生一组唯一的序列号。
- **数据库链路**是指一个数据库与另一个数据库之间的通信路径。
- **快照**是对远程数据库上表的复制，自动按时间间隔定时刷新表的数据。



9.2 PL/SQL程序设计

- 1、过程化SQL- PL / SQL
- 是Oracle对标准数据库语言的过程化扩展。
- 2、PL/SQL程序的基本结构
- (1)PL/SQL块结构
- DECLARE <说明的变量、常量、游标等> //声明部分
- BEGIN <SQL语句、PL / SQL的流程控制语句> //执行部分
- [EXCEPTION <异常处理部分>] //异常处理部分
- END;

PL/SQL操作符分为：算术、关系、比较及逻辑操作符，与其他程序设计语言类同。

- (2)变量类型与赋值
- 分为数字型、布尔型、字符型和日期型。
- (1) 直接给变量赋值：<变量名> := <表达式>
- (2)通过查询语句给变量赋值：

3. 控制结构

- **主要有条件控制和循环控制语句：**
- （1) 条件控制语句
- 条件结构有三种形式：IF—THEN，IF—THEN—ELSE和嵌套的IF语句。
- **常用:** IF <条件> THEN <语句1>；ELSE <语句2> END IF；
- （2) 循环控制语句
- 有三种循环结构，提供EXIT、BREAK等循环结束语句。形式：
- ① LOOP <循环体> END LOOP;
- ② WHILE <条件> LOOP <循环体> END LOOP;
- ③ FOR <循环变量> IN <下界> .. <上界>
  LOOP <循环体> END LOOP

3. 控制结构

- 例：通过循环变量i来控制n增加的次数并输出结果。
- SET SERVEROUTPUT ON
- DECLARE
  n INTEGER :=80； i INTEGER :=0；
- BEGIN
  FOR i IN 1..10
  LOOP n:=n+1；
  END LOOP；
  DBMS_OUTPUT.PUT_LINE('n的值：'||TO_CHAR(n))
- END；

4. 存储过程

- Oracle提供了四种类型的可存储的命名程序块：
  存储过程、函数、包和触发器。
- 存储过程是一个PL/SQL程序块，可有多个参数作为输入或输出源。它不能由SQL语句直接使用，只能通过EXECUT命令或PL/SQL程序块内部调用。
- 可通过SQL语句创建、重命名、执行和删除存储过程。
- （1）创建存储过程
- CREATEP OR REPLACE PROCEDURE <过程名>
  [(<参数1>[,<参数2>,...])]　//过程首部
- AS　<PL / SQL块>;　//存储过程体
- （2）调用存储过程
- CALL PROCEDURE<过程名>([实参1，实参2…])

存储过程举例：

- 设：有员工表；部门表。

emp表

eno	ename	salary	dno
7369	王利	3000	2

dept表

dno	dname	tel
2	销售	68779880

- (1)建表并输入数据
- ① 创建员工表：
- CREATE TABLE emp
  (eno NUMBER(5), ename VARCHAR2(20), salary NUMBER(4),dno NUMBER(1),
  PRIMARY KEY (eno));
- ② 创建部门表：……
- ③ 给emp表添加记录。存储过程如下：
- CREATE OR REPLACE PROCEDURE ins_table_emp
  (e_id NUMBER,e_name VARCHAR2,e_sal NUMBER,e_dno NUMBER)
  AS BEGIN
  INSERT INTO emp(eno,ename,salary, dno)
  VALUES(e_id,e_name,e_sal,e_dno);
  END;

(2)操作相关表，使得某部门职工的薪水增加15%。

- 给指定部门的员工加薪，首先要把该部门的员工选出来，再对这些员工的薪水进行改动。
- 这里用存储过程表示，将要加薪的部门作为参数：

```
CREATE OR REPLACE PROCEDURE add_sal (deptname varchar2)
AS   BEGIN
UPDATE emp
SET emp.salary=emp.salary*1.15
WHERE emp.eno IN
     (SELECT eno
     FROM dept
     WHERE dname= deptname);
END ;
```

(3)利用触发器建立追踪

- 通过对emp表的salary属性创建一个触发器，来监视其更新并进行记录，以追踪薪水变动情况：
- CREATE OR REPLACE TRIGGER salary_change
  BEFORE DELETE OR INSERT OR UPDATE ON emp　--触发事件
  FOR EACH ROW　　-- 每更新一行都需要调用此触发器
- DECLARE --只有触发器的声明需要DECLARE，过程和函数都不要

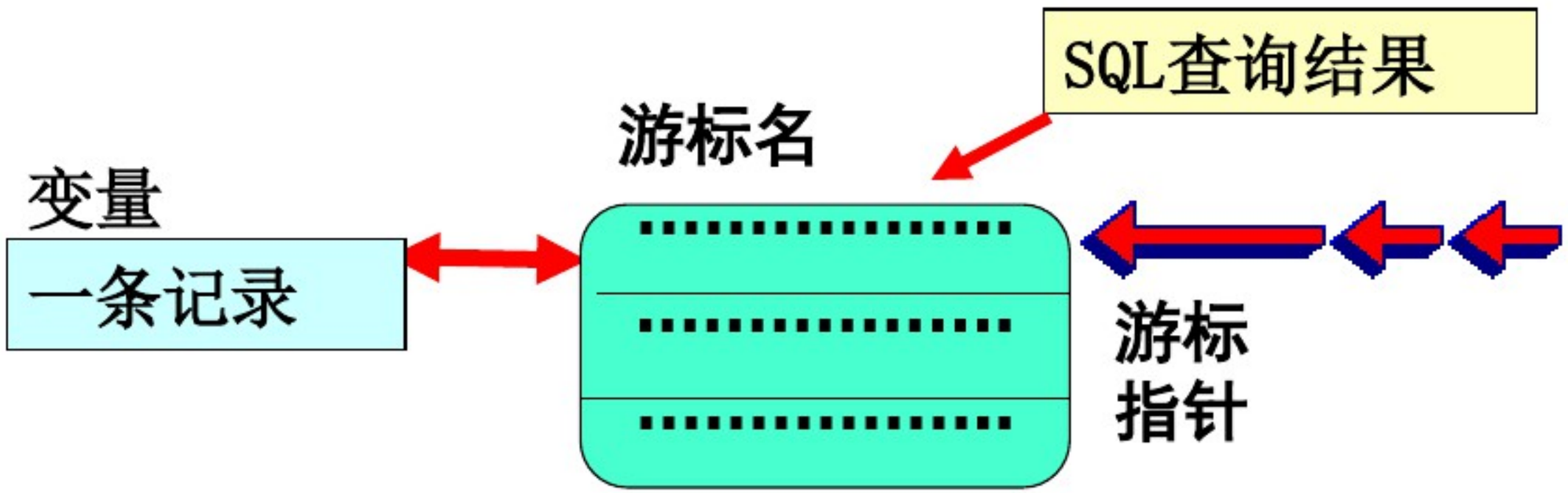
```
salary1 NUMBER;
BEGIN    --:new与:old分别代表该行在修改前、改后的记录
salary1=:new.salary - old.salary;
DBMS_OUTPUT.PUT_LINE('old salary is:'|| :old.salary);
DBMS_OUTPUT.PUT_LINE('new salary is:'|| :new.salary);
DBMS_OUTPUT.PUT_LINE('add is:'||to_char(salary1));
END ;
```

5、游标的使用与设计

游标：是系统开设的一个数据缓冲区，存放SQL语句的执行结果。

- **作用：用户可通过游标获取记录，并赋给变量。**

- 当对数据库的查询操作返回一组结果集时，存入游标，以后通过对游标的操作来获取结果集中的数据信息。
- 游标分：显式游标和隐式游标。当查询语句返回多条记录时，必须显式地定义游标以处理每一行。其他的SQL语句(更新操作或查询操作只返回一条记录)都使用隐式游标。



5、游标的使用与设计

- **定义游标：** CURSOR <游标名> IS <SQL语句>;
- 例：CURSOR c_emp IS
  SELECT * FROM emp WHERE dno=3;
- 当需要操作结果集时，须完成：打开游标、使用FETCH语句将游标里的数据取出以及关闭游标操作。
- **游标的四个属性:**其值代表游标的状态信息。游标属性的值是由系统自动赋值，用户只能使用，不能改变。

游标属性	描　　述
<游标名>%ISOPEN	如果游标已打开，取值为TRUE，否则为FALSE
<游标名>%NOTFOUND	若最近一次FETCH操作无结果,则值为TRUE,否则为FALSE
<游标名>%FOUND	若最近一次FETCH操作无结果,则值为FALSE,否则为TRUE
<游标名>%ROWCOUNT	值是到当前为止返回的记录数，初值为0， 每取一条记录，该属性值加1

为emp表的员工增加10%的工资,总额限制在50万元以内。

```
CURSOR addsal IS
SELECT eno, salary FROM emp  ORDER BY salary
FOR UPDATE OF salary;
  emp_num  number:= 0;                    --声明员工计数变量
  s_sal   emp.salary%TYPE;                --声明员工总工资变量
  e_sal   emp.salary%TYPE;                --声明员工工资变量
  e_no    emp.eno%TYPE;                   --声明员工号变量
BEGIN OPEN addsal;
      SELECT sum(salary) INTO s_sal FROM emp;
      WHILE s_sal< 500000
      LOOP       FETCH addsal INTO e_no, e_sal;
                 EXIT WHEN addsal % NOTFOUND;
    UPDATE emp SET salary=salary*1.1 WHERE CURRENT OF addsal;
           s_sal:=s_sal + e_sal*0.1; emp_num:=emp_num + 1 ;
       END LOOP;
CLOSE addsal;
INSERT INTO msg VALUES(emp_num , s_sal) ; --将结果存入表msg
COMMIT; END;
```

PL/SQL程序块的区别

程序块	描　　述	应用环境
存储过程、函数	可接受参数并返回结果的PL / SQL块，存储在服务器端,可被重复调用	任何客户端和服务器环境
包	有名PL / SQL块，是相关过程、函数、标识符的集合，存储在服务器端，可被重复调用。	任何客户端和服务器环境
数据库触发器	与数据库表相关的PL / SQL块，存储在服务器端，在客户与服务器端的触发事件发生时自动触发。	任何客户端和服务器环境
应用触发器	与一个应用事件相关的PL / SQL块，存储在应用程序中，在应用程序的触发事件发生时自动触发。	各种工具

9.3 数据库应用系统设计。

- 9.3.1 系统需求与设计目标
- 9.3.2 系统功能模块设计
- 9.3.3 系统实现方式及开发环境
- 9.3.4 数据库设计与表的创建
- 9.3.5 系统应用程序设计
- 9.3.6 系统测试运行图

DB应用系统设计例—见教材

第8章-第9章数据库应用 小结

- **数据库应用开发的一般过程是:**
- (1) 进行需求分析，确定应用系统的功能需求和性能需求。
- (2) 根据需求分析的结果划分系统的功能模块，并进行数据库各种结构设计。
- (3) 建立数据库与创建数据库连接。
- (4) 应用程序设计与编码(包括界面)的实现。
- (5) 装入数据、测试及维护。