# Plsql基本概念

## 1、背景

[结构化查询语言](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E6%9F%A5%E8%AF%A2%E8%AF%AD%E8%A8%80" \t "/home/vincent/Documents\\x/_blank)(Structured Query Language，简称SQL)是用来访问关系型数据库一种通用语言，属于[第四代语言](https://baike.baidu.com/item/%E7%AC%AC%E5%9B%9B%E4%BB%A3%E8%AF%AD%E8%A8%80" \t "/home/vincent/Documents\\x/_blank)（4GL），其执行特点是非过程化，即不用指明执行的具体方法和途径，而是简单地调用相应语句来直接取得结果即可。显然，这种不关注任何实现细节的语言对于开发者来说有着极大的便利。然而，有些复杂的业务流程要求相应的程序来描述，这种情况下4GL就有些无能为力了。PL/SQL的出现正是为了解决这一问题，它是一种过程化语言，属于[第三代语言](https://baike.baidu.com/item/%E7%AC%AC%E4%B8%89%E4%BB%A3%E8%AF%AD%E8%A8%80" \t "/home/vincent/Documents\\x/_blank)，它与C、 C++、Java等语言一样关注于处理细节，可以用来实现比较复杂的业务逻辑。

PL/SQL是[Oracle数据库](https://baike.baidu.com/item/Oracle%E6%95%B0%E6%8D%AE%E5%BA%93" \t "/home/vincent/Documents\\x/_blank)对SQL语句的扩展。在普通SQL语句的使用上增加了编程语言的特点，所以PL/SQL把数据操作和查询语句组织在PL/SQL代码的过程性单元中，通过逻辑判断、循环等操作实现复杂的功能或者计算。

因为PL/SQL允许混合SQL声明和过程结构，因此可以在将声明发送到甲骨文系统去执行之前使用PL/SQL区段和副程序来组合SQL声明，没有PL/SQL，甲骨文需要就每次处理SQL声明，在网络环境中，这将影响交通流量，而且增加响应时间。PL/SQL区段只被[编译](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91" \t "/home/vincent/Documents\\x/_blank)一次并且以可运行的形式储存，以降低响应时间。

## 2、plsql程序结构

PL/SQL程序都是以块（block）为基本单位，整个PL/SQL块分三部分：声明部分（用declare开头）、执行部分（以 begin开头）和[异常处理](https://baike.baidu.com/item/%E5%BC%82%E5%B8%B8%E5%A4%84%E7%90%86" \t "/home/vincent/Documents\\x/_blank)部分（以exception开头）。其中执行部分是必须的，其他两个部分可选。无论PL/SQL[程序段](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E6%AE%B5" \t "/home/vincent/Documents\\x/_blank)的代码量有多大，其基本结构就是由这三部分组成。

*declare  
　　/\* 声明区(可选)：定义类型和变量、声明变量、声明函数、游标 \*/  
　　begin  
　　/\* 执行区(必须的)：执行pl/sql语句或者sql语句 \*/  
　　exception  
　　/\* 异常处理区(可选)：处理错误的 \*/  
　　end;*

*如下所示为一段完整的PL/SQL块：*

*/\*声明部分，以declare开头\*/*

*declare v\_id integer;*

*v\_name varchar(20);*

*cursor c\_emp is select \* from employee where emp\_id=3;*

*/\*执行部分，以begin开头\*/*

*begin 　open c\_emp;　//打开[游标](https://baike.baidu.com/item/%E6%B8%B8%E6%A0%87" \t "/home/vincent/Documents\\x/_blank)*

*loop*

*fetch c\_emp into v\_id,v\_name;　 　//从游标取数据*

*exit when c\_emp%notfound ;*

*end loop ;*

*close c\_emp;　 　 //关闭游标*

*dbms\_output.PUT\_LINE(v\_name);*

*/\*[异常处理](https://baike.baidu.com/item/%E5%BC%82%E5%B8%B8%E5%A4%84%E7%90%86" \t "/home/vincent/Documents\\x/_blank)部分，以exception开始\*/*

*exception*

*when no\_data\_found then*

*dbms\_output.PUT\_LINE('没有数据');*

*end ;*

## plsql控制结构

PL/SQL[程序段](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E6%AE%B5" \t "/home/vincent/Documents\\x/_blank)中有三种程序结构：条件结构、[循环结构](https://baike.baidu.com/item/%E5%BE%AA%E7%8E%AF%E7%BB%93%E6%9E%84" \t "/home/vincent/Documents\\x/_blank)和[顺序结构](https://baike.baidu.com/item/%E9%A1%BA%E5%BA%8F%E7%BB%93%E6%9E%84" \t "/home/vincent/Documents\\x/_blank)。

1) 条件结构

与其它语言完全类似，语法结构如下：

*if condition then*

*statement1*

*else*

*statement2*

*end if ;*

2)循环结构

这一结构与其他语言不太一样，在PL/SQL程序中有三种循环结构：

*a. loop … end loop;*

*b. while condition loop … end loop;*

*c. for variable in low\_bound . .upper\_bound loop … end loop;*

其中的“…”代表循环体。

3)[顺序结构](https://baike.baidu.com/item/%E9%A1%BA%E5%BA%8F%E7%BB%93%E6%9E%84" \t "/home/vincent/Documents\\x/_blank)

实际就是goto的运用，不过从程序控制的角度来看，尽量少用goto可以使得程序结构更加的清晰。

## 4、plsql变量生命与赋值

PL/SQL主要用于数据库编程，所以其所有的数据类型跟[Oracle数据库](https://baike.baidu.com/item/Oracle%E6%95%B0%E6%8D%AE%E5%BA%93" \t "/home/vincent/Documents\\x/_blank)里的字段类型是一一对应的，大体分为数字型、[布尔型](https://baike.baidu.com/item/%E5%B8%83%E5%B0%94%E5%9E%8B" \t "/home/vincent/Documents\\x/_blank)、字符型和日期型。这里简单介绍两种常用数据类型：number、varchar2。

number

用来存储整数和浮点数。范围为1e130～10e125，其使用语法为：

number[(precision, scale)]

其中(precision, scale)是可选的，precision表示所有数字的个数，scale表示小数点右边数字的个数。

varchar2

用来存储变长的字符串，其使用语法为：

varchar2[(size)]

其中size为可选，表示该字符串所能存储的最大长度。

在PL/SQL中声明变量与其他语言不太一样，它采用从右往左的方式声明，比如声明一个number类型的变量v\_id，那其形式应为：

v\_id number;

如果给上面的v\_id变量赋值，不能用“=”，应该用“:=”,即形式为：

v\_id :=5;

## 5、plsql基本命令

PL/SQL使用的数据库操作语言还是基于SQL的，所以熟悉SQL是进行PL/SQL编程的基础。SQL语言的分类情况大致如下：

1.数据定义语言（DDL），包括CREATE（创建）命令、ALTER（修改）命令、DROP（删除）命令等。

2. 数据操纵语言（DML），包括INSERT（插入）命令、UPDATE（更新）命令、DELETE（删除）命令、SELECT …FOR UPDATE（查询）等。

3. 数据查询语言（DQL），包括基本查询语句、Order By子句、Group By子句等。

4. 事务控制语言（TCL），包括COMMIT（提交）命令、SAVEPOINT（保存点）命令、ROLLBACK（回滚）命令。

5. 数据控制语言（DCL），GRANT（授权）命令、REVOKE（撤销）命令。

具体的语法结构可以参阅其他关于SQL语言的资料，这里不再赘述。

## 6、过程和函数

PL/SQL中的过程和函数与其他语言的过程和函数一样，都是为了执行一定的任务而组合在一起的语句。过程无返回值，函数有返回值。其语法结构为：

过程：Create or replace procedure procname(参数列表) as PL/SQL语句块

函数：Create or replace function funcname(参数列表) return 返回值 as PL/SQL语句块

为便于理解，举例如下：

问题：假设有一张表t1，有f1和f2两个字段，f1为number类型，f2为varchar2类型，要往t1里写两条记录，内容自定。

Create or replace procedure test\_procedure as

V\_f11 number :=1; /\*声明变量并赋初值\*/

V\_f12 number :=2;

V\_f21 varchar2(20) :='first';

V\_f22 varchar2(20) :='second';

Begin

Insert into t1 values (V\_f11, V\_f21);

Insert into t1 values (V\_f12, V\_f22);

End test\_procedure; /\*test\_procedure可以省略\*/

至此，test\_procedure[存储过程](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E8%BF%87%E7%A8%8B" \t "/home/vincent/Documents\\x/_blank)已经完成，经过[编译](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91" \t "/home/vincent/Documents\\x/_blank)后就可以在其他PL/SQL块或者过程中调用了。函数与过程具有很大的相似性，此处不再详述。

## plsql游标

[游标](https://baike.baidu.com/item/%E6%B8%B8%E6%A0%87" \t "/home/vincent/Documents\\x/_blank)的定义为：用游标来指代一个DML SQL操作返回的[结果集](https://baike.baidu.com/item/%E7%BB%93%E6%9E%9C%E9%9B%86" \t "/home/vincent/Documents\\x/_blank)。即当一个对数据库的查询操作返回一组结果集时，用游标来标注这组结果集，以后通过对游标的操作来获取结果集中的数据信息。这里特别提出游标的概念，是因为它在PL/SQL的编程中非常的重要。定义游标的语法结构如下：

cursor cursor\_name is SQL语句;

在本文第一段代码中有一句话如下：

*cursor c\_emp is select \* from employee where emp\_id=3;*

其含义是定义一个游标c\_emp，代表employee表中所有emp\_id字段为3的[结果集](https://baike.baidu.com/item/%E7%BB%93%E6%9E%9C%E9%9B%86" \t "/home/vincent/Documents\\x/_blank)。当需要操作该结果集时，必须完成三步：打开[游标](https://baike.baidu.com/item/%E6%B8%B8%E6%A0%87" \t "/home/vincent/Documents\\x/_blank)、使用fetch语句将游标里的数据取出、关闭游标。

游标用来处理从数据库中检索的多行记录（使用SELECT语句）。利用游标，程序可以逐个地处理和遍历一次检索返回的整个[记录集](https://baike.baidu.com/item/%E8%AE%B0%E5%BD%95%E9%9B%86" \t "/home/vincent/Documents\\x/_blank)。

为了处理SQL语句，Oracle将在内存中分配一个区域，这就是上下文区。这个区包含了已经处理完的行数、指向被分析语句的[指针](https://baike.baidu.com/item/%E6%8C%87%E9%92%88" \t "/home/vincent/Documents\\x/_blank)，整个区是查询语句返回的数据行集。游标就是指向上下文区句柄或指针。

### 显式游标

显示[游标](https://baike.baidu.com/item/%E6%B8%B8%E6%A0%87" \t "/home/vincent/Documents\\x/_blank)被用于处理返回多行数据的SELECT 语句，游标名通过CURSOR….IS 语句显示地赋给SELECT 语句。

在PL/SQL中处理显示游标所必需的四个步骤：

1）声明游标；CURSOR cursor\_name IS select\_statement

2）为查询打开游标；OPEN cursor\_name

3）取得结果放入PL/SQL变量中；

*FETCH cursor\_name INTO list\_of\_variables;*

*FETCH cursor\_name INTO PL/SQL\_record;*

4）关闭[游标](https://baike.baidu.com/item/%E6%B8%B8%E6%A0%87" \t "/home/vincent/Documents\\x/_blank)。CLOSE cursor\_name

注意：在声明游标时，select\_statement不能包含INTO子句。当使用显示游标时，INTO子句是FETCH语句的一部分。

### 隐式游标

所有的隐式游标都被假设为只返回一条记录。

使用[隐式游标](https://baike.baidu.com/item/%E9%9A%90%E5%BC%8F%E6%B8%B8%E6%A0%87" \t "/home/vincent/Documents\\x/_blank)时，用户无需进行声明、打开及关闭。PL/SQL隐含地打开、处理，然后关掉[游标](https://baike.baidu.com/item/%E6%B8%B8%E6%A0%87" \t "/home/vincent/Documents\\x/_blank)。

例如：

*SELECT studentNo,studentName*

*INTO curStudentNo,curStudentName*

*FROM StudentRecord*

*WHERE name=’gg’;*

上述游标自动打开，并把相关值赋给对应变量，然后关闭。执行完后，PL/SQL变量curStudentNo,curStudentName中已经有了值。