# 1 Oracle

## 1.1基本概念

要了解ORACLE数据库，需要先了解以下基本概念：

数据库：是一个数据集合，我们大多数情况下讲的数据库概念不仅是指物理数据，还指内存、进程对象的组合。ORACLE数据库都将数据存储在文件中，在其内部，数据库结构提供了数据对文件的逻辑映射，允许不同类型的数据分开存放。这些逻辑划分称为表空间。关于表空间的概念将在下文中介绍。

* 表空间（Tablespace）：是数据库的逻辑划分，每个数据库至少有一个表空间（system表空间）。
* 数据文件（DataFile）：每个表空间由同一磁盘上的一个或多个文件组成，这些文件叫数据文件。建立新表空间需要建立新的数据文件。
* 实例（Instance）：也称为服务器（server），是存取和控制数据库的软件机制，它由系统全局区SGA和后台进程组成。

ORACLE数据库启动时，实际是启动ORACLE实例，（安装并打开数据库）。一个数据库可以被多个实例访问（这是ORACLE的并行服务器选项）。决定实例的大小及组成的参数存储在init.ora文件中，实例启动时需要读这个文件，并且在运行时可以由数据库管理员修改，但要在下次启动实例时才会起作用。

## 1.2 SQL语言分类

* 数据查询语言：SELECT语句，用于检索数据库数据。
* 数据操纵语言（DML）：用于改变数据库数据，包括INSERT、DELETE、UPDATE语句。
* 事务控制语言（TCL）：用于维护数据的一致性，包括COMMIT、ROLLBACK、SAVEPOINT语句，COMMIT提交事务，ROLLBACK回滚事务，SAVEPOINT用于设置保存点以取消部分数据的改变。
* 数据定义语言（DDL）：用于建立、修改和删除数据库对象。CREATE TABLE建表，ALTER TABLE 修改表结构，DROP TABLE删除表。注意：DDL语句会自动提交事务。
* 数据控制语言（DCL）：用于执行权限授予和回收操作，包括GRANT和REVOKE命令。注意：DCL语句会自动提交事务。

## 1.3 SQL开发工具

### 1.3.1 SQL\*Plus

在命令行运行SQL\*Plus是使用sqlplus命令来完成的，语法如下：

sqlplus [username]/[password]@[server]

当连接到本地数据库时，不需要提供网络服务名[server]；当在客户端连使用SQL\*Plus接远程数据库时，必须使用Net Manager配置网络服务名，并且使用该网络服务名连接到远程的数据库。

### 1.3.2 PL/SQL Developer

PL/SQL Developer是用于开发PL/SQL子程序的集成开发环境（IDE），它是一个独立的产品（收费的）。

# 2 PL/SQL

## 2.1 PL/SQL块

PL/SQL块由三个部分组成：

* 定义部分：用于定义常量、变量、游标、例外、复杂数据类型等；
* 执行部分：用于实现应用模块功能，包含了要执行的PL/SQL语句和SQL；
* 例外处理部分：用于处理执行部分可能出现的运行错误。

PL/SQL块结构如下：

DECLARE

/\*

\*定义部分

\*/

BEGIN

/\*

\* 执行部分：PL/SQL 和 SQL语句

\*/

EXCEPTION

/\*

\* 例外处理部分

\*/

END; /\* 块结束标记 \*/

其中，定义部分以DECLARE开始，该部分可选；执行部分以BEGIN开始，该部分是必须的；例外处理部分以EXCEPTION开始，是可选的；而END则是PL/SQL块结束标志。注意：DECLARE、BEGIN、EXCEPTION后面都没有分号（;），而 END后面必须带分号。

### 2.1.1 PL/SQL块分类

根据需要实际的应用模块功能，PL/SQL块划分为匿名块、命名块、子程序和触发器四种类型。

* 匿名块：指没有名称的PL/SQL块，匿名块可内嵌到应用程序中(JAVA、C++)，也可以在交互式环境中直接使用（SQL\*PLUS）
* 命名块：指具有特定名称的PL/SQL块，在PL/SQL块前使用<<>>加以标志
* 子程序：包括存储过程、函数和包

存储过程：可指定输入参数(IN)，也可以指定输出参数(OUT)。

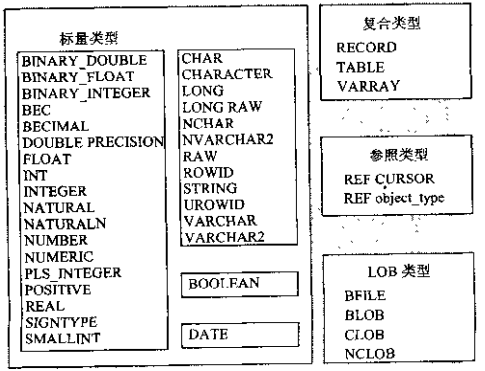
函数：用于返回特定数据。

包：用于逻辑组合过程和函数 ，由包规范和包体组成。

* 触发器：指隐含执行的存储过程。必须指定触发事件及触发操作，常用的触发事件包括INSERT、UPDATE、DELETE语句，触发的实际是PL/SQL块

## 2.2 变量

在编写PL/SQL程序时，若临时存储数值，必须定义变量、常量、和参数；且必须指定数据类型。PL/SQL中可使用标量类型、复合类型、参照类型和LOB(Large Object)类型，如下图



### 2.2.1 标量变量

标量变量是指只能存放单个数值的变量。标量数据类型包括数字类型、字符类型、日期类型和布尔类型。常用的有：

* VARCHAR2(n)：定义可变长度的字符串，n为最大长度，最大值为 32767字节。在PL/SQL块中，VARCHAR2长度不应超过4000字节。
* CHAR(n)：固定长度的字符串
* NUMBER(p,s)：固定长度的整数和浮点数
* DATE：定义日期和时间数据
* BOOLEAN：定义布尔变量

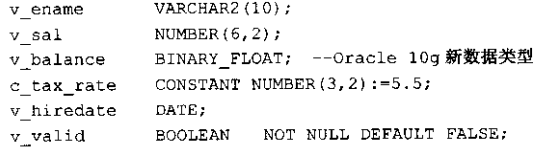
**定义标量变量**

必须首先定义标量变量，然后才能执行部分和例外部分中使用标量变量。语法：

identifier [CONSTANT] datatype [NOT NULL] [:= |DEFAULT expr]

* identifier：用于指定常量或变量的名称。
* CONSTANT：指定常量。定义常量时，必须指定它的初始值，并且不能改变。
* datatype：数据类型
* NOT NULL：用于强制初始化变量（不能为NULL），此时必须为变量提供数值。
* := ：用于为变量和常量指定初始值
* DEFAULT：为变量和常量指定初始值
* expr：用于指定初始值的PL/SQL表达式，可以是文本值、函数、其他变量

如果没有指定初始值，那么变量的初始值为NULL。示例如下：



**使用标量变量**

在定义部分定义变量后，在执行部分和例外处理部分可以使用标量变量。在PL/SQL中在等号前加冒号表示赋值：“:=”

**使用%TYPE属性**

使用%TYPE属性定义变量时，它会按照数据库列或其他变量来确定新变量的类型和长度。示例如下：

DECLARE

v\_name emp.ename%TYPE;

BEGIN

..

END;

### 2.2.2 复合变量

复合变量指用于存放多个值的变量。当定义复合变量时，需使用PL/SQL的复合数据。如：PL/SQL记录、PL/SQL表、嵌套表等

### 2.2.3 参照变量

参照变量是指用于存放数值指针的变量。通过使用参照变量，可以使得应用程序共享相同对象，降低占用空间。可使用游标变量(REF CURSOR)和对象类型变量等参照变量类型。

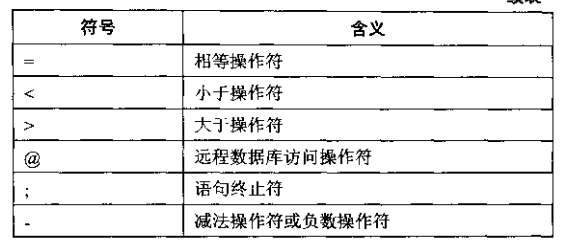
### 2.2.4 LOB变量

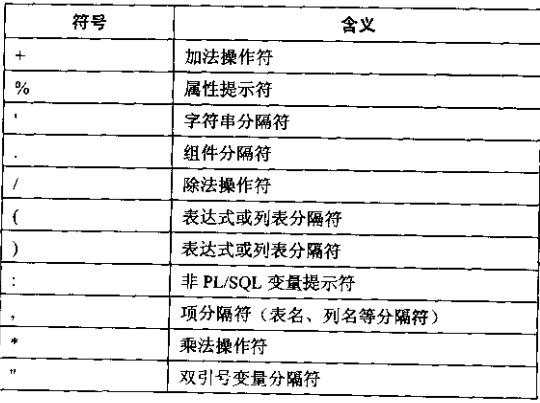
LOB变量指用于存储大批量数据的变量

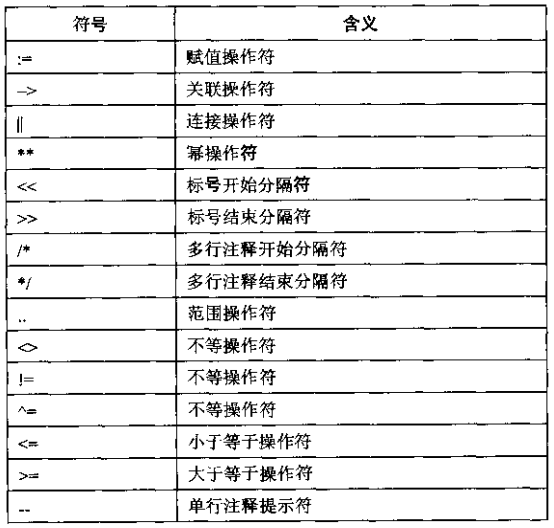
## 2.3 编写PL/SQL 代码

### 2.3.1 词汇单元

* 分隔符：指具有特定含义的单个符号或组合符号。







## 控制结构

### 条件分支语句

PL/SQL提供三种条件语句：IF-THEN, IF-THEN-ELSE, IF-THEN-ELSIF。语法如下：

IF condition THEN

statements;

[ELSIF condition THEN

statements;]

[ELSE condition THEN

statements;]

END IF;

### CASE语句

**单一条件**

当使用CASE语句执行多重条件分支时，如果条件选择符完全相同，并且条件表达式为相等条件选择，使用单一选择符进行等值比较。语法如下：

CASE selector –条件选择符

WHEN expression1 THEN statements1; --条件值的表达式和要执行的条件操作

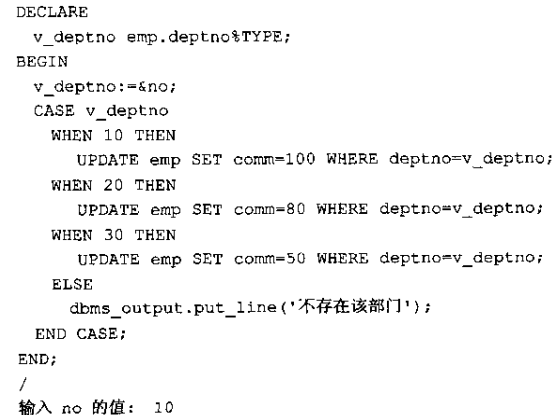
WHEN expression2 THEN statements2;

…

WHEN expressionN THEN statementsN;

END CASE;

示例如下：



**多种条件比较**

如果包含有多种条件进行不等比较，那么必须在WHEN子句中指定比较条件。语法如下：

CASE

WHEN search\_condition1 THEN statements1;--不同的比较条件和满足条件执行的操作

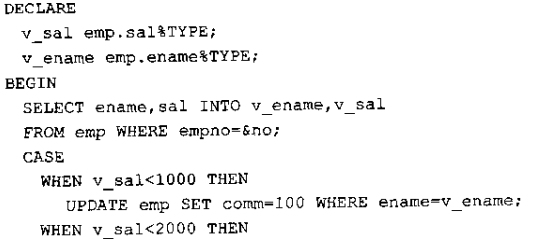
WHEN search\_condition2 THEN statements2;

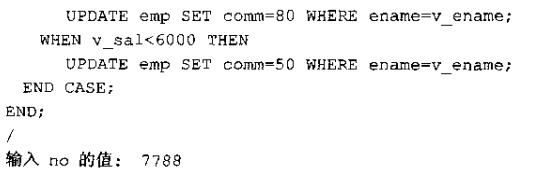
…

WHEN search\_conditionN THEN statementsN;

END CASE;

示例如下：





### 循环语句

编写PL/SQL块中重复执行一条语句或者一组语句，可使用循环结构。

**基本循环**

基本循环至少要执行一次循环体内的语句。以LOOP开始，以END LOOP结束。语法如下：

LOOP

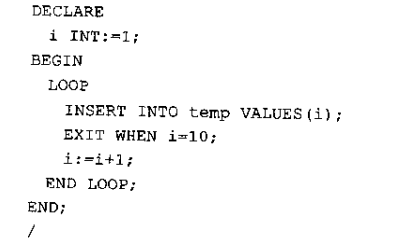
statement1;

….

EXIT [WHEN condition]

END LOOP;

示例如下，使用基本循环为TEMP表插入10条数据，并且当i=10时退出循环。



**WHILE循环**

对于while循环来说，只有条件为TRUE时，才会执行循环体内的语句。以WHILE…LOOP开始，END LOOP结束。语法如下：

WHILE condition LOOP

statement1;

statement2;

…

END LOOP;

**FOR 循环**

使用FOR循环时，Oracle会隐含定义循环控制变量。语法如下：

FOR counter in [REVERSE] –counter循环控制变量，隐含定义

lower\_bound…upper\_bound LOOP –循环变量的下界值和上界值

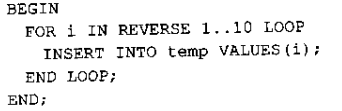
statement1;

statement2;

…

END LOOP;

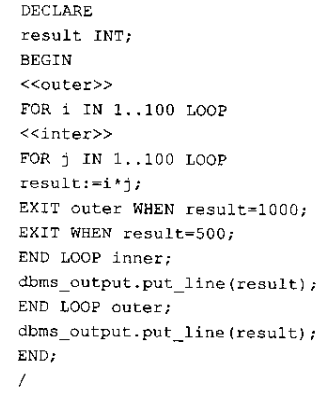
示例如下：



**嵌套循环和标号**

嵌套循环是指一个循环语句中嵌入另外一个循环语句，而标号则用于标记嵌套块或嵌套循环。通过标号，可以区分内循环和外层循环，在内循环中直接退出外层循环。

示例如下

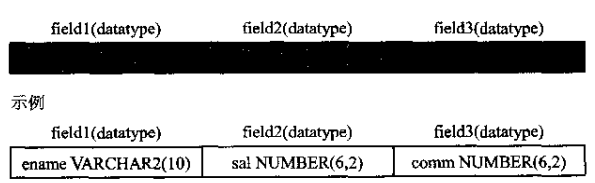


如果result=1000，直接退出外层循环，而当result=5000时只会退出内层循环。

## 复合数据类型

### PL/SQL记录

PL/SQL记录(Record)类似于高级语言中的结构， 方便处理单行多列数据。它由一组相关的记录成员（field）组成。结构如下：



**定义**

使用PL/SQL记录，可自定义记录类型和记录变量，也可使用%ROWTYPE属性直接定义记录变量。

* 自定义PL/SQL记录，语法如下：

TYPE type\_name IS RECORD(

field\_declaration[,

field\_declaration]…

);

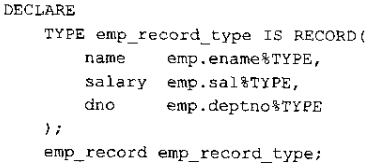
identifier type\_name;

type\_name：记录类型

field\_declaration：记录成员的定义

identifier：指定记录变量名

示例如下：



* 使用%ROWTYPE定义记录，可基于表或视图定义记录变量。记录成员的名称和类型与表或视图列的名称和类型完全相同。语法如下：

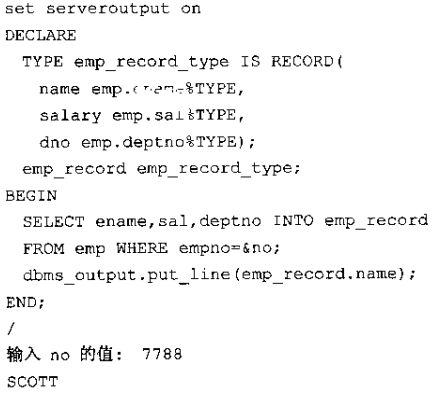
identifier table\_name%ROWTYPE;

identifier view\_name%ROWTYPE;

**使用PL/SQL记录**

* 在SELECT INTO语句中使用PL/SQL记录

SELECT INTO用于检索单行数据。当在SELECT INTO语句中直接使用记录变量时，选择列表中的列和表达式的顺序、个数、类型必须要与记录成员的顺序、个数、类型完全匹配。示例如下：



### PL/SQL集合

处理单列多行的数据，可使用PL/SQL集合。集合类型包括索引表、嵌套表、变长数组。