**一、PL/SQL基本语法**

1、定义常量：<常量名> constant <数据类型> :=<值>；

2、定义变量：<变量名> <数据类型> [(宽度) ：= <初始值>]；

3、赋值符号为“：=”

4、条件判断语句

（1）IF语句：

IF condition THEN

statements;

[ELSEIF condition THEN

statements;]

[ELSE

statements;]

END IF;

（2）CASE语句：

CASE case\_operand

WHEN when\_operand THEN

statement;

[WHEN when\_operand THEN

statement;

]…

[ELSE statement [statement]]…;

END CASE;

5、循环语句

（1）简单循环

LOOP

statement1;

…

EXIT [WHEN condition];

END LOOP;

（2）FOR循环

FOR index\_name IN [REVERSE]

lower\_bound…upper\_bound LOOP

statement1;

statement2;

…

END LOOP;

（3）WHILE循环

WHILE condition LOOP

statement1;

statement2;

…

END LOOP;

1. **常用函数**
2. ABS函数用于返回给定数字表达式的绝对值。
3. ROUND函数返回数字表达式并四舍五入为指定的长度或精度。
4. CEIL函数返回大于或等于所给数字表达式的最小整数。
5. FLOOR函数返回小于或等于所给数字表达式的最大整数。
6. POWER函数返回给定表达式指定次方的值。
7. LENGTH函数返回给定字符串表达式的字符（而不是字节）个数。
8. UPPER函数返回将小写字符数据转换为大写的字符表达式。LOWER函数返回将大写字符数据转换为小写的字符表达式。
9. REPLACE函数具体的语法结构是REPLACE(s1,s2,s3)，表示在字符串s1中搜索字符串s2，并用字符串s3替换s2。
10. RPAD函数的语法结构是RPAD(s1,n[,s2])，功能是在字符串s1的右边用字符串s2填充，直到整个字符串长度为n时为止。如果s2不存在，则以空格填充。
11. LPAD函数的语法结构是LPAD(s1,n[,s2])，功能是在字符串s1的左边用字符串s2填充，直到整个字符串长度为n时为止。
12. TRIM函数具体的语法结构是TRIM(p,s)，功能是将删除指定的前导或尾随的字符，默认删除空格。LTRIM函数从字符串s左侧开始删除。RTRIM函数从字符串s右侧开始删除。
13. SYSDATE函数返回当前日期和时间。SYSTIMESTAMP函数没有参数，用来返回系统时间，该时间包含时区信息，精确到微秒。
14. EXTRACT (datetime)该函数可以从指定的时间当中提取到指定的日期部分。
15. COUNT函数语法结构是：COUNT(\*|[distinct][all]exp)，该函数用来计算记录的数量或某列的个数。函数中必须指定列名，或使用星号代表统计所有记录的个数。
16. TO\_CHAR函数转换数值和日期为字符串。
17. TO\_DATE函数可将字符型数据转换成日期型数据。
18. CAST函数语法结构是：CAST(expr as type\_name)，该函数实现了数据类型之间的转换。

**控制日志文件**

控制文件是一个很小的二进制文件，它包含了数据库的结构信息（即构成数据库的数据文件和日志文件）。**主要包含**：数据库名称、数据文件和重做日志文件的位置和名称、数据库创建的时间戳、当前日志序列号、检查点信息、表空间名称、备份和恢复信息。**作用**：控制文件用来描述Oracle实体结构，在启动Oracle数据库时读取控制文件才能取得所有数据库文件的相关信息

**重做日志文件**

重做日志文件由重做记录组成，而每个重做记录由一组变化元素组成，变化元素中记录了数据库中每个单独的数据块的变化情况。**作用**：提高数据库的性能。即使发生故障导致数据库崩溃，Oracle也可以利用重做信息恢复丢失的数据。

**归档日志**

在重做日志文件被覆盖之前，Oracle能够将已经写满的重做日志文件通过复制保存到指定的位置，保存下来的所有重做日志文件被称为“归档重做日志”，这个过程就是“归档过程”。 **作用**：解决重做日志文件被新写入的重做日志文件覆盖的问题。使用归档日志文件可以实现恢复数据库、更新备用数据库以及获取数据库的历史信息等目的。

**索引**

**概念：**索引是以表的列为基础的数据库对象，它保存着表中排序的索引项，并且记录索引列在数据表中的物理存储位置，实现表中数据的逻辑排序

**功能：**一是为了实施主键和唯一性约束，再者是为了提高性能。

**使用索引代价：**1、索引需要占用数据表以外的物理存储空间。2、创建和维护索引要花费一定的时间。3、当对表进行更新操作时，索引需要被重建，这样就降低了数据的维护速度。

**创建索引的原则：**1、根据表的大小来创建索引2、根据列的特征来创建索引3、在一个表上创建索引的数量不是越多越好

**视图**

**概念：**视图是一种数据库对象，它是从一个或多个表或视图导出的虚表。视图是从基表中导出的逻辑表，它不像基表一样物理地存储在数据库中，视图没有自己独立的数据实体。一个视图的存在反映在数据字典中具有相应的登记项。视图一旦被建立，即可在其上进行DML操作。

**优点：**1.简化用户操作2.视图使用户以多角度看待同一数据3.视图对重构数据库提供了一定程度的逻辑独立性4.视图可增加安全性

**作用**：为用户以多种角度观察数据库中的数据提供方便。  
 **目的**：用于集中、简化和定制显示数据库中的数据信息

**使用视图的作用主要表现在以下几个方面：**1、提供面向用户的数据表现形式2、  
提供面向用户的安全性保证3、隐藏数据的逻辑复杂性4、简化用户权限的管理5、重构数据库的灵活性

**序列**

**概念：**序列是一种数据库对象，用来自动产生一组唯一的序号。序列是一种共享式的对象，多个用户可以共同使用序列中的序号。

**优点：**一般序列所生成的整数通常可以用来填充数字类型的主键列，这样当向表中插入数据时，主键列就使用了序列中的序号，从而保证主键的列值不会重复。用这种方法可以替代在应用程序中产生主键值的方法，可以获得更可靠的主键值。

**存储过程**

存储过程是Oracle开发者在数据转换或查询报表时最经常使用的方式之一。存储过程是一种命名PL/SQL程序块，它将一些相关的SQL语句，流程控制语句组合在一起，用于执行某些特定的操作或者任务，可以将经常需要执行的特定的操作写成过程。通过过程名，就可以多次调用过程，从而实现程序的模块化设计。这种方式极大地节省了用户的时间，也提高了程序的效率。

**优点：**（1）存储过程在服务器端运行，执行速度快。（2）存储过程执行一次后驻留在Oracle数据库服务器的高速Cache中，以后再次执行存储过程时，只需从高速Cache中调用已经编译好的代码即可，从而提高了系统性能。（3）存储过程确保了数据库的安全。使用存储过程，可以在禁止用户直接访问应用程序中的某些数据表的情况下，授权执行访问这些数据表的存储过程。（4）自动完成需要预先执行的任务。存储过程可以设置为系统启动时自动执行，而不必在系统启动后再进行手动操作，从而方便了用户的使用，可以自动完成一些需要预先执行的任务。

**函数 ：**函数与过程很相似，它也是存储在数据库中的命名程序块。函数与存储过程的区别在于，函数必须有返回值，过程可以没有；它只能作为一个表达式的一部分，不能作为一个独立的语句来使用。

**包 ：**包也称之为程序包，是指相关的过程、函数、变量、常量和游标等PL/SQL程序设计元素的组合。

**触发器 ：**触发器是一种特殊的存储过程，它与数据表紧密联系，用于保护表中的数据，当一个定义了特定类型触发器的基表执行插入、修改或删除表中数据的操作时，将自动触发触发器中定义的操作，以实现数据的一致性和完整性。

**作用：**（1）在安全性方面，触发器可以基于数据库的值使用户具有操作数据库的某种权利。 （2）在审计方面，触发器可以跟踪用户对数据库的操作。（3）实现复杂的数据完整性规则。 （4）实现复杂的非标准的数据库相关完整性规则。触发器可以对数据库中相关的表进行连环更新。（5）同步实时地复制表中的数据。（6）自动计算数据值，如果数据的值达到了一定的要求，则进行特定的处理。

**事务 ：**事务在数据库中主要用于保证数据的一致性，防止出现错误数据。在事务内的语句都会被看成一个单元，一旦有一个失败，那么所有的都会失败。在编程过程中也经常用到事务。

**特性：**1、原子性（Atomicity）：事务是一个完整的操作。事务的各步操作是不可分的（原子的）；要么都执行，要么都不执行2、一致性（Consistency）：事务把数据库从一个一致性状态带入另一个一致性状态。3、隔离性（Isolation）：对数据进行修改的所有并发事务是彼此隔离的，这表明事务必须是独立的，它不应以任何方式依赖于或影响其他事务。4、永久性（Durability）：事务完成后，它对数据库的修改永久有效，事务日志能够保持事务的永久性。

事务开始于第一条SQL语句，在下列之一情况结束：遇到COMMIT或ROLLBACK命令；遇到一条DDL或者DCL命令；系统发生错误、退出或者崩溃。

控制事务的方式有两种：  
**1. 隐式控制** 语法格式如下：**SET AUTOCOMMIT ON/OFF**

设置开关为ON，表示自动提交数据更新语句；设置开关为OFF，表示关闭自动提交数据更新语句的功能

**2. 显式控制** 显式方式就是利用COMMIT和ROLLBACK命令“显式”的结束事务。

Oracle中的事务不需要设置开始标志，通常遇到登录数据库后，第一次执行DML语句的时候；或者是当事务结束后，第一次执行DML语句的时候，事务就开始了。

在事务中可以根据用户的需要设置保存点（在其所在位置之前的事务语句不能回滚的位置）

使用SAVEPOINT命令可以设置事务**保存点，其基本语法格式如下：SAVEPOINT <保存点名>;**

**锁 ：**加锁是实现数据库并发控制的一个非常重要的技术。当事务在对某个数据对象进行操作前，先向系统发出请求，对其加锁。加锁后事务就对该数据对象有了一定的控制，在该事务释放锁之前，其他的事务不能对此数据对象进行更新操作。Oracle通过使用锁机制维护数据的完整性、并发性和一致性。

**锁的类型：**1、DML锁（Data Locks，数据锁）：用于保护数据的完整性；DML锁主要包括TM锁和TX锁，其中TM锁称为表级锁，TX锁称为事务锁或行级锁。 2、DDL锁（Dictionary Locks，字典锁）：用于保护数据库对象的结构，如表、索引等的结构定义。 3、内部锁和闩（Internal Locks and Latches）：保护数据库的内部结构。

**系统全局区：**

**概念：**当数据库服务器上的一个数据库启动时，Oracle将分配一块内存区间，叫做系统全局区（SGA）。一个实例只有一个SGA。SGA中的数据可以在多个用户进程之间共享。SGA由若干个缓存和缓冲池组成

**SGA包含以下几个数据结构：**  
**1. 数据库缓冲区高速缓存（缓冲区高速缓存是SGA的最大组成部分。）**

**作用：**用于存储用户最近使用过的数据

检索时数据库内的数据先读取到该区域，然后再经过处理后返回给用户 ；修改时被修改的数据先被存储在数据库缓冲区内，而不是马上写入磁盘（该数据块称为脏缓存块）。   
**优点**：可以减少磁盘读写次数，提高系统的存取效率，改善系统性能。  
**2. 重做日志缓冲区**

**作用对象**：重做日志是对用户事务所产生的记录 **作用**：通过重做日志能够重新产生数据

重做日志缓冲区是一个圆形缓冲区，其大小由初始化参数LOG\_BUFFER指定。重做日志缓冲区越大，就可以记录越多的用户操作，写重做日志文件的次数也就越少  
**优点**：可以提高数据库的性能 **限制条件**：重做日志缓冲区的大小是受物理内存大小的限制的。  
**3. 共享池**

**目的**：为了提高数据库的性能 **作用**：用于存放与SQL语句的执行有关的信息  
**组成：**由三部分组成，数据字典缓存、库缓存和服务器结果缓存  
**4. Java池（是SGA中一段可选的存储区域）**

**作用**：用来存放运行Java所必需的共享代码和共享数据  
Java池的大小由初始化参数JAVA\_POOL\_SIZE指定，默认大小是20MB，大小是动态的，可以自动管理。

1. **大池**

**概念**：是数据库管理员配置的可选内存区域 **作用**：用于分配大量的内存  
**应用场景**：在共享服务器连接、语句并行执行或者备份的时候都可用到大池。

**表空间**

是Oracle数据库中最大的逻辑存储单位，同时也是数据库的存储空间单位，系统通过表空间存储模式对象。

Oracle使用表空间将相关的逻辑结构组合在一起，每个表空间由一个或多个数据文件组成，但是一个数据文件只能属于一个表空间，这是逻辑与物理的统一。所以，数据库的存储空间在物理上体现为数据文件，而在逻辑上表示为表空间。

根据表空间对盘区的管理方式，表空间可以分为数据字典管理的表空间和本地化管理的表空间两种。

**安全性管理：**数据库访问的安全性主要包括两个方面：1、阻止为授权用户访问数据库2、每个数据库用户都有不同的操作权限

**系统权限：**指数据库级别执行某些操作的权限，即用户执行某一特定的数据库操作或某类数据库操作等的权限，例如创建表空间、创建会话等。

填空：

1. 创建数据库的语句是**CREATE DATABASE**。
2. 执行立即关闭的命令是**SHUTDOWN IMMEDIATE**。
3. 删除数据库的命令是**DROP DATABASE**。
4. 启动数据库的步骤是**启动实例、装载数据库、打开数据库**。
5. 执行强制启动数据库的命令是 **STARTUP FORCE**。
6. 在ALTER TABLE语句中，删除列的关键字是 **DROP COLUMN**。
7. 在ALTER TABLE语句中，重命名表的关键字是**RENAME TO**。
8. 在ALTER TABLE语句中，删除约束的关键字是 **DROP CONSTRAINT**。
9. 在T-SQL语句中**SELECT**语句使用频率最高。
10. 左外连接返回连接中左表的**全部**数据行，而只返回右表中**部分**数据行。
11. SELECT查询语句中两个必不可少的子句是**SELECT**和**FROM**。
12. 显示游标的处理包括**声明游标**、**打开游标**、**提取游标**、**关闭游标**四个步骤。
13. **DECLARE**关键字标志着PL/SQL程序中声明段的开始，在声明段中可以声明变量、常量和游标等对象。
14. 返回小于或等于所给数字表达式的最大整数的函数是**FLOOR**。
15. 返回当前日期和时间的函数是**SYSDATE**。
16. 引入索引的目的是**加快查询速度**。
17. 序列号是一个Oracle整数，最多可有**38**个数字。
18. SQL Plus中使用**EXECUTE**命令来执行存储过程。
19. **SYSDATE**函数可以获得当前系统的日期。
20. **CREATE FUNCTION**语句可以用来创建函数。
21. **DROP TRIGGER**语句可以删除触发器。
22. **:OLD**变量表用于存储DELETE和UPDATE语句所影响的行的值。
23. **UPDATE**操作既会影响:NEW变量和:OLD变量。
24. Oracle中使用**COMMIT**命令提交事务。
25. Oracle中使用**ROLLBACK**命令回滚事务。
26. Oracle中使用**SAVEPOINT**命令设置保存点。
27. 锁被分成 **共享锁、排他锁** 两种基本类型。
28. 查看表空间数据文件信息的视图是**dba\_data\_files**。
29. 创建临时表空间的关键字是**CREATE TEMPORARY TABLESPACE**
30. 创建大文件表空间的语法格式是**CREATE BIGFILE TABLESPACE**。
31. 删除表空间的语法格式是**DROP TABLESPACE。**
32. 查看临时表空间的数据字典是**DBA\_TEMP\_FILES**
33. 创建撤销表空间的关键字是**CREATE UNDO TABLESPACE**。

**Oracle中function和procedure的区别？**

1. 可以理解函数是存储过程的一种

2. 函数可以没有参数,但是一定需要一个返回值，存储过程可以没有参数,不需要返回值

3. 函数return返回值没有返回参数模式，存储过程通过out参数返回值, 如果需要返回多个参数则建议使用存储过程

4. 在sql数据操纵语句中只能调用函数而不能调用存储过程

**创建一个简单的存储过程Insert\_student，该过程可以用于向学生表添加数据。**

SQL>CREATE OR REPLACE PROCEDURE Insert\_student AS  
 BEGIN  
 INSERT INTO Students  
 VALUES(‘1514010726’,’张馨’,’女’,to\_date(‘1997-10-12’,’YYYY-MM-DD),’汉族’,’081’,’08101’);  
 EXCEPTION//异常情况处理  
 WHEN DUP\_VAL\_ON\_INDEX THEN//查重 重复就自动执行update  
 DBMS\_OUTPUT.PUT\_LINE(‘重复的学生ID’);  
 WHEN OTHERS THEN//如果有其他的工作  
 DBMS\_OUTPUT.PUT\_LINE(‘发生其他错误！’);  
 END Insert\_student;

**查看存储过程Insert\_student的脚本，代码如下。**

SQL> SELECT \* FROM USER\_SOURCE WHERE NAME='Insert\_student' ORDER BY LINE; **创建一个带有输入参数的存储过程：当使用Inseret\_student添加学生信息时，用户只需要向该存储过程传入参数值，然后由存储过程从中读取数据。**

SQL>CREATE OR REPLACE PROCEDURE Insert\_student

（P\_ID IN VARCHAR2,

P\_NAME IN VARCHAR2, P\_SEX IN VARCHAR2, P\_BIRTHDAY IN DATE,

P\_NAT IN VARCHAR2, P\_DID IN VARCHAR2, P\_CID IN VARCHAR2）

IS

BEGIN

INSERT INTO Students

VALUES(P\_ID, P\_NAME, P\_SEX, P\_BIRTHDAY, P\_NAT, P\_DID, P\_CID);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN DBMS\_OUTPUT.PUT\_LINE('重复的学生ID');

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('发生其他错误！');

END Insert\_student;

**使用IN OUT参数交换两个变量的值，代码如下**

CREATE OR REPLACE PROCEDURE P\_SWAP(p\_v1 in out number, p\_v2 in out number)

IS

v\_temp number;

BEGIN

v\_temp := p\_v1;

p\_v1 := p\_v2;

p\_v2 := v\_temp;

END P\_SWAP;

DECLARE

Var\_1 number:=10;

Var\_2 number:=20;

BEGIN

P\_SWAP(var\_1,var\_2);

DBMS\_OUTPUT.PUT\_LINE('var\_1='||var\_1);

DBMS\_OUTPUT.PUT\_LINE('var\_2='||var\_2);

END;

**创建一个带有输出参数的存储过程：该存储过程根据学生的姓名返回学生该生选修的所有课程的课程名。**

**(1)使用SQL Developer工具创建存储过程**

CREATE OR REPLACE PROCEDURE P\_COURSENAME(P\_SNAME IN VARCHAR2,

COURSENAME OUT VARCHAR2)

IS

V\_COURSENAME VARCHAR2(50);

BEGIN

FOR STUDENT IN (SELECT C.CNAME

FROM STUDENTS S, SC , COURSES C

WHERE S.STUDENT\_ID = SC.STUDENT\_ID

AND SC.COURSE\_ID = C.COURSE\_ID

AND S.SNAME =P\_SNAME) LOOP

V\_COURSENAME := V\_COURSENAME || STUDENT.CNAME ;

END LOOP;

COURSENAME := V\_COURSENAME;

END P\_COURSENAME;

**(2)执行存储过程**

VARIABLE C\_NAME VARCHAR2(20)

EXEC P\_COURSENAME('王传璐',:C\_NAME);

PRINT C\_NAME;

**创建一个带有默认值的存储过程：该存储过程根据学生的学号返回该生选修的所有课程的总学分。**

**(1)创建存储过程**

SET SERVEROUTPUT ON

CREATE OR REPLACE PROCEDURE sum\_course\_credit

(stud\_id IN varchar2 DEFAULT '20110810101')

AS

sum\_credit NUMBER;

BEGIN

SELECT sum(Credits) INTO sum\_credit

FROM Courses

WHERE course\_id in(SELECT course\_id

FROM sc

WHERE Student\_id=stud\_id);

DBMS\_OUTPUT.PUTLINE(sum\_credit);

END sum\_course\_credit;

**(2)执行存储过程**

BEGIN

sum\_course\_credit(' 20110810203');

sum\_course\_credit();

END;

**创建存储过程INSERT\_TEA：该过程用于向教师表插入信息时确保输入的教师工资不小于2000元。**

--修改教师表，添加工资列

ALTER TABLE TEACHER

ADD SAL NUMBER DEFAULT 2000;

--创建存储过程

CREATE OR REPLACE PROCEDURE INSERT\_TEA(

TID IN VARCHAR2,

TName IN VARCHAR2,

TSex IN VARCHAR2 DEFAULT '男',

TBirth IN DATE,

TEduc IN VARCHAR2 DEFAULT '研究生',

TPro IN VARCHAR2 DEFAULT '讲师',

TDeptID IN VARCHAR2,

TSAL IN NUMBER DEFAULT '2000') IS

t\_sal teacher.sal%TYPE;

PROCEDURE check\_sal(salary in out number) IS

BEGIN

IF salary< 2000 THEN salary:=2000;

END IF;

END check\_sal;

BEGIN

t\_sal:=TSAL;

check\_sal(t\_sal);

INSERT INTO

TEACHER(Teacher\_ID,Tname,Tsex,TBirth,Educ,Profess,Dept\_ID,SAL)

VALUES (TID,TNAME,TSex,TBirth,TEduc,TPro,TDeptID,t\_sal);

END INSERT\_TEA;

**删除例116-创建的存储过程，代码如下：**

DROP PROCDURE sum\_course\_credit;

**创建函数用于求两个数的平均值。 使用SQL Plus工具创建该函数，名称为F\_AVG**

SQL>CREATE OR REPLACE FUNCTION F\_AVG

( p\_num1 IN NUMBER

, p\_num2 IN NUMBER  
 ) RETURN NUMBER AS

RESULT\_AVG NUMBER;

BEGIN

RESULT\_AVG:=(p\_num1+p\_num2)/2;

RETURN (RESULT\_AVG);

END F\_AVG;

**下面的匿名程序块中调用函数F\_AVG（）获取两个数的平均值：**SQL> SET SERVEROUTPUT ON

SQL>BEGIN   
DBMS\_OUTPUT.PUT\_LINE(F\_AVG(88,99)); END;

**创建标准索引：**CREATE INDEX 索引名 ON 表名 (列名) TABLESPACE 表空间名;

**创建唯一索引:**CREATE unique INDEX 索引名 ON 表名 (列名) TABLESPACE 表空间名;

**创建组合索引:**CREATE INDEX 索引名 ON 表名 (列名1,列名2) TABLESPACE 表空间名;

**创建反向键索引:**CREATE INDEX 索引名 ON 表名 (列名) reverse TABLESPACE 表空间名;

**创建函数用于计算选修某门课程的学生人数.**

CREATE OR REPLACE FUNCTION COUNT\_STU(F\_CNAME IN VARCHAR2)

RETURN NUMBER;

AS

RESULT NUMBER := 1;

BEGIN

SELECT COUNT(DISTINCT STUDENT\_ID) INTO RESULT

FROM SC, COURSES C

WHERE COURSE\_ID = C.COURSE\_ID

AND C.CNAME = F\_CNAME;

RETURN(RESULT);

END COUNT\_STU;

**创建函数的基本语法如下：**CREATE [OR REPLACE] FUNCTION function\_name

[ parameter1 [IN | OUT | IN OUT ]data\_type1,

parameter2 [ IN | OUT | IN OUT ] data \_type2,

……]

RETURN data\_type;

IS|AS

声明部分

BEGIN

执行部分

EXCEPTION

异常处理部分

END [function\_name];