六、常用单行函数

大多数关系型数据库都支持SQL语句，但为了方便对数据的操作，每个数据库都有自己的的函数库，

Oracle函数分为单行函数和多行函数。所谓单行函数就是能够对每一条记录进行操作并且返回一条

结果与之对应，单行函数可以修改字段数据类型，可以嵌套使用；常用单行函数有如下5种：

* 字符函数
* 数值函数
* 日期函数
* 转换函数
* 通用函数

单行函数有如下几种常用功能：

* 对数据进行计算；
* 控制数据的输出格式；
* 设置/改变日期的显示格式；
* 进行数据类型转换；
* 使用NVL 函数处理空值；
* 实现IF-THEN-ELSE 多路分支逻辑；

# 第一节 字符函数

字符函数负责对字符串数据进行操作。常见字符函数如下：

|  |
| --- |
| UPPER(字符串 | 字段)：将字符串中字母全部转换为大写； |
| LOWER(字符串 |字段)：将字符串中的字母全部转换为小写； |
| INITCAP(字符串 |字段)：使首字母大写； |
| LENGTH(字符串 |字段)：求出字符串长度； |
| REPLACE(字符串 |字段，替换字符，被替换字符)：进行字符替换； |
| SUBSTR(字符串 |字段，开始点 [，结束点])：字符串截取； |

**补充：**

**a)对字符进行大小写转换**

|  |
| --- |
|  |

**b)对字符串进行处理**

|  |
| --- |
|  |

**注：**在oracle中，要使用单行函数，需要提供完整的SQL语句，故oracle提供了一个虚拟表dual用来

充当辅助表，使得SQL语句完整。

1. 大写转换函数的使用；

|  |
| --- |
|  |

**注：**由于Oracle中的数据区分大小写，故操作时需要注意大小写，使用大小写转换函数可

以方便地将数据转换成符合规定的数据，而用户不必花太多时间在意函数大小写。

若不使用大小写转换函数可能出现以下问题：

|  |
| --- |
|  |

1. 查询emp表，查找指定输入的字符串；

|  |
| --- |
|  |

**注：**&input 表示代换变量，从控制台读取相应的参数进行代换，input为自定义参数名。

1. 小写转换函数的使用，查询emp表，将所有员工姓名以小写字母返回；

|  |
| --- |
|  |

1. 首字母大写转换函数的使用，查询emp表，将雇员姓名首字母大写；

|  |
| --- |
|  |

1. 字符串长度函数的使用，查询emp表，将返回雇员姓名长度；

|  |
| --- |
|  |

1. 字符串长度函数的使用，查询emp表，将返回雇员姓名长度为5的雇员姓名；

|  |
| --- |
|  |

1. **字符替换函数的使用，查询emp表，将员工姓名中所有‘A’用‘\_’替换；**

|  |
| --- |
|  |

1. 字符串截取函数的使用有两种语法：

* SUBSTR(字符串 | 列，开始点 ，结束点)：截取指定长度字符串；
* SUBSTR(字符串 | 列，开始点)：截取从开始点到结束的字符串；

**注**：为了方便用户使用，oracle做了下标重构智能，下标0和下标1等效；

|  |
| --- |
|  |

1. 字符串截取函数的使用，查询emp表，截取员工姓名后3位返回；

思路一：用员工姓名长度减2；

思路二：用负数表示从字符串后边开始截取的开始点和结束点；

|  |
| --- |
|  |

# 第二节 数字函数

|  |
| --- |
| ROUND(数字|字段[，保留小数的位数]):将数字四舍五入； |
| TRUNC(数字|字段[，保留小数的位数]):舍弃指定位之后的内容，并不四舍五入； |
| MOD(数字1，数字2):取余；（取模） |

**补充：**

|  |
| --- |
|  |

1. ROUND函数的使用；

|  |
| --- |
|  |
|  |
|  |

**注：**round(数字，负数)表示从整数位开始四舍五入；

1. TRUNC函数的使用；

|  |
| --- |
|  |

1. MOD函数的使用；

|  |
| --- |
|  |

# 第三节 日期函数

Oracle内部以数字格式存储日期和时间信息：世纪,年,月,日,小时,分钟,秒；

## 3.1获取当前日期

要想对日期进行操作，首先应先获取当前日期；获取当前日期用‘SYSDATE’；

1. 获取当前日期；

|  |
| --- |
|  |

**注**：日期还可以进行四则运算；有以下三个公式：

1. 日期 + 数字 = 日期：表示若干天后的日期；
2. 日期 - 数字 = 日期：表示若干天前的日期；
3. 日期 - 日期 = 数字：表示两日期相隔的天数
4. 日期的加减运算

|  |
| --- |
|  |

## 3.2 常用日期操作函数

|  |
| --- |
| LAST\_DAY(日期)：求出指定日期所在月份的最后一天； |
| NEXT\_DAY(日期，星期数或是星期字符串)：求出下一个指定星期数的日期； |
| ADD\_MONTHS(日期，数字)：求出若干个月之后的日期； |
| MONTHS\_BETWEEN(日期1，日期2)：求出两个日期之间所经历的月份； |

**补充：**

|  |
| --- |
|  |

1. 求出本月的最后一天；

|  |
| --- |
|  |

1. 求出下一个星期一是几号；

|  |
| --- |
|  |

**注：**1表示的是星期天，2表示星期一,0在此无意义；

|  |
| --- |
|  |

**注：**亦可用星期字符串表示星期；如下：

|  |
| --- |
|  |

1. 查询四个月之后的日期；

|  |
| --- |
|  |

1. 查询emp表，查出员工从雇佣到现在工作的月数并取整；

|  |
| --- |
|  |

**注**：负数表示第一个日期在第二个日期之前，调换两日期位置即可；

**注：**在对数据中日期进行操作时，可以使用数字来表示日期，具体算法可以在程序中定义，但建议使用日期函数对日期进行操作，这样可以避免闰年带来的不便；

# 第四节 转换函数

现在已经用到了Oracle的三种数据类型：数字型NUMBER，字符串型VARCHAR2,日期型DATE,转换函数

的目的就是完成Oracle数据类型之间的转换。Oracle中数据类型转换包括隐含转换和显式转换两种方式，

建议使用显式的数据类型转换，确保SQL语句的可靠性。转换函数转换用法如下图：

|  |
| --- |
|  |

|  |
| --- |
| TO\_CHAR(字符串|字段，格式字符串)：将日期或者数字按照规定的格式转为字符串显示； |
| TO\_DATE(字符串，格式字符串)：将字符串变为规定格式的日期显示； |
| TO\_NUMBER(字符串)：将字符串转变为数字显示； |

**补充：**

1. **常用日期格式符：**

|  |
| --- |
|  |

**注：**除上述格式符外，日期模式串中还可直接出现如下字符**- : ; /**，如要显示其它文本字符串则

需使用双引号括起来；也可在模式串的开头使用"fm"标记以去掉数字前面的零。

1. **数字模式符**：

|  |
| --- |
|  |

## 4.1 TO\_CHAR函数

TO\_CHAR函数有两个用法：1）日期格式化；2）数字格式化；格式化是按照格式字符串参数来

确定相应格式的；常见格式化参数及意义：

|  |  |
| --- | --- |
| yyyy | 表示年份 |
| mm | 表示月份 |
| dd | 表示日期 |
| hh | 表示小时 |
| mi | 表示分钟 |
| ss | 表示秒 |
| fm | 去掉前导0，例如月份08变为8 |
| L | Local,表示当前所在语言环境的货币单位 |
| 9 | 表示一位数字，可以是0-9的任意一个 |

1. 日期格式化；

在默认情况下日期格式如下：

|  |
| --- |
|  |

这不太符合我们的习惯，不便于阅读和理解，故可以将其格式化为我们所习惯的表示方式：

|  |
| --- |
|  |

还可以根据格式字符串拆分日期：

|  |
| --- |
|  |

**注：**DATE数据附带时间，但默认并没有显示，可以通过格式字符串读取时间；

|  |
| --- |
|  |

在hh后加24即可改为24时计时法：

|  |
| --- |
|  |

在格式字符串前加fm即可去掉前导0，但不常用：

|  |
| --- |
|  |

1. 数字格式化；

|  |
| --- |
|  |

**注：**TO\_DATE函数格式化后的数据都是字符串类型的；

## 4. 2 TO\_DATE函数

TO\_DATE函数，将字符串表示的日期转换为日期格式。

1. TO\_DATE函数的使用；

|  |
| --- |
|  |

此函数在程序与数据库交互时使用较多。

## 4.3 TO\_NUMBER函数

将字符串表示的数字转化为数字格式的数字。

1. TO\_NUMBER函数的使用；

|  |
| --- |
|  |

**注**：Oracle中已经有机制可默认将字符串表示的数字看做数字格式的数字，故此函数并不常用。

|  |
| --- |
|  |

# 第五节 特色函数

Oracle中有两个特色函数，方便对数据的处理：

|  |
| --- |
| NVL(字段，替换null的值)：将字段中null替换为指定的值； |
| DECODE(数值|字段，判断值1，显示值1，判断值2，显示值2 [,判断值3，显示值3…]):根据对数值或字段的判断，替换显示相应的内容； |

**补充：**

|  |  |  |
| --- | --- | --- |
| **nvl ()** | 函数用于将空值null替换为指定的缺省值，适用于  字符、数字、日期等类型数据  **语法格式：**   |  | | --- | | **NVL(exp1, exp2)** | |
| **nvl2 ()** | NVL2()函数用于实现条件表达式功能。  **语法格式：**   |  | | --- | | **NVL2(exp1, exp2, exp3)** |   如果表达式exp1的值不为null，则返回exp2的值，  否则返回exp3的值。 |
| **nullif ()** | 函数用于数据等价性比较并根据比较结果返回null  或其中一个被比较的数值。  **语法格式：**   |  | | --- | | **nullif(exp1, exp2)** |   如果表达式exp1与exp2的值相等则返回null，否则  返回exp1的值。 |
| **coalesce ()** | 依次考察各参数表达式，遇到非null值即停止并返  回该值。  **语法格式：**   |  | | --- | | **coalesce(exp1, exp2, …)** | |
| **case表达式** | 表达式用于实现多路分支结构；  语法格式：   |  | | --- | | **case exp when comparison\_exp1 then return\_exp1**  **[when comparison\_exp2 then return\_exp2**  **when comparison\_expn then return\_expn**  **else else\_exp]**  **end** | |
| **decode()** | decode()函数也用于实现多路分支结构；  **语法格式：**   |  | | --- | | **decode(col|expression, search1, result1**  **[, search2, result2,...,]**  **[, default])** | |

## 5.1 NVL函数

NVL函数用来对数据中的null值进行替换，使其符合相关格式或友好显示。

1. NVL函数的使用，查询emp表，查出雇员的年薪记录；

|  |
| --- |
|  |

从中可以看出，当comm为空时，年薪无法计算，也为空，故需要将comm中为空的数据替换为0，从而符合要求；

|  |
| --- |
|  |

## 5.2 DECODE函数

DECODE函数类似于if语句，不同点是DECODE函数判断的是数值而非逻辑条件。

1. 查询emp表，查询雇员职位并用中文显示；

emp表中，job中的类型和对应中文：

|  |  |
| --- | --- |
| CLERK | 职员 |
| SALESMAN | 销售人员 |
| MANAGER | 经理 |
| ANALYST | 分析员 |
| PRESIDENT | 董事长 |

|  |
| --- |
|  |

**注：熟记scott中的四张表结构。**

# 第六节 系统函数

**sys\_context()函数：**用于查看用户数据库信息；

* terminal:当前会话客户所对应的终端的标志符
* language:语言
* db\_name:当前数据库名称
* nls\_date\_format:当前会话客户所对应的日期格式
* session\_user:当前会话客户所对应的数据库用户名
* current\_schema:当前会话客户所对应的默认方案名；
* host:返回数据库所在的主机名称；

|  |
| --- |
| SELECT SYS\_CONTEXT('userenv','db\_name') FROM dual; |
|  |