# 一、decode()函数

**1、decode(**条件,值1,返回值1,值2,返回值2,...值n,返回值n,缺省值)

该函数的含义如下：

IF 条件=值1 THEN

　　　　RETURN(翻译值1)

ELSIF 条件=值2 THEN

　　　　RETURN(翻译值2)

　　　　......

ELSIF 条件=值n THEN

　　　　RETURN(翻译值n)

ELSE

　　　　RETURN(缺省值)

END IF

**decode**(字段或字段的运算，值1，值2，值3）

       这个函数运行的结果是，当字段或字段的运算的值等于值1时，该函数返回值2，否则返回值3 。 值1，值2，值3也可以是表达式，

# 二、Round() 函数

格式如下：ROUND（number[,decimals]）

其中：number 待做截取处理的数值

decimals 指明需保留小数点后面的位数。可选项，忽略它则截去所有的小数部分，并四舍五入。如果为负数则表示从小数点开始左边的位数，相应整数数字用0填充，小数被去掉。需要注意的是，和trunc函数不同，对截取的数字要四舍五入。

# 三、Nvl（）函数

NVL()是Oracle PL/SQL中的一个函数。它的格式是NVL( string1, replace\_with)。它的功能是如果string1为NULL，则NVL函数返回replace\_with的值，否则返回string1的值，如果两个参数都为NULL ，则返回NULL。

NVL2 (expr1, expr2, expr3) ->expr1不为NULL，返回expr2；为NULL，返回expr3。expr2和expr3类型不同的话，expr3会转换为expr2的类型

NULLIF (expr1, expr2) ->相等返回NULL，不等返回expr1

# 四、case when then else

--简单Case函数

CASE sex

WHEN '1' THEN '男'

WHEN '2' THEN '女'

ELSE '其他' END

--Case搜索函数

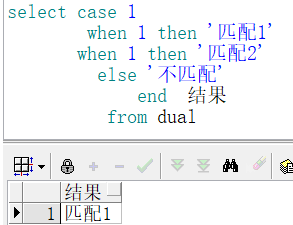
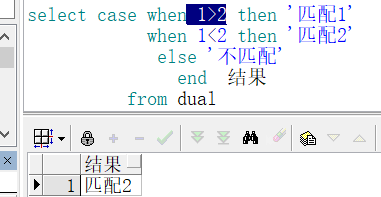
CASE WHEN sex = '1' THEN '男'

WHEN sex = '2' THEN '女'

ELSE '其他' END

Case when…then …else…end，也相当于一系列 if…else..的嵌套使用

|  |
| --- |
| Case when ‘条件’ then ‘条件为真的值’ else ‘条件为假的值’ end ；  只返回第一个条件满足的值，配合sum() 和group by 可以分组统计的效果 |

# 五、substr()函数

功能：截取字符串

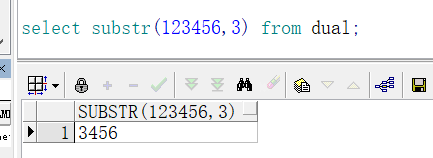
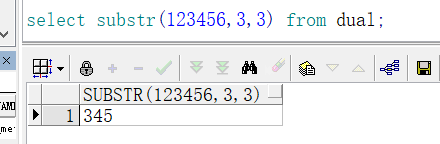
函数格式substr(‘原字符串’，‘截取的位置’，[截取的长度])

原字符串：即需要截取的字符串

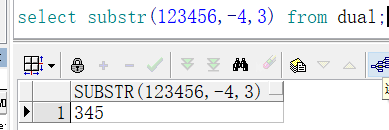
截取的位置：字符串开始截取的位置，可以为正数和负数，正数时：从字符串左边开始数。为负数时：从字符串右边开始数；（0和1都是从第一个位置开始）

[截取的长度]：从截取位置开始，需要截取多少个字符。可以省略：截取位置之后的所有字符。

省略截取长度： 添加略截取长度：

截取位置为负数



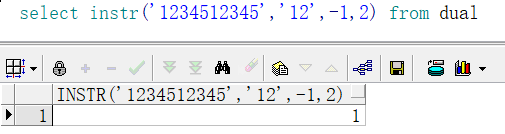
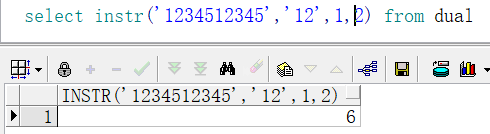
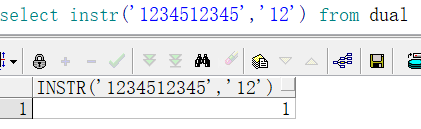
# 六、Instr（）函数

功能：查找子串在原字符中的位置

格式：Instr(‘原字符串’，‘字串’，[开始匹配位置]，[第几个匹配项])

[开始匹配位置]可省略：默认从左第一个位置开始匹配，为负数时，从右向左开始匹配

[第几个匹配项]可省略：默认返回第一个配的位置



# 七、|| 连接字符==concat()

区别：|| 可以一次连接多个字符串，

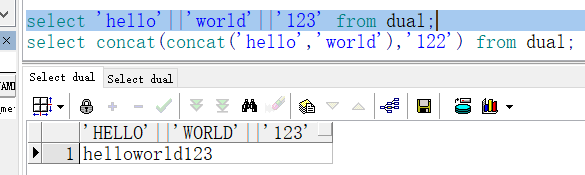
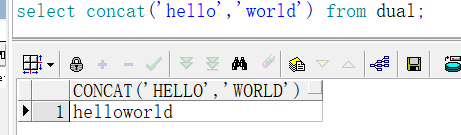
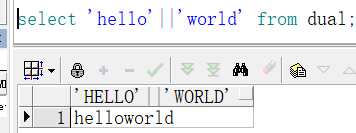
trim(),去除前后空格，

length（）求字符的长

lower（）字符变小写

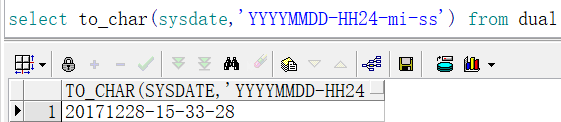
upper（）字符变大写

initcap（）首字母大写



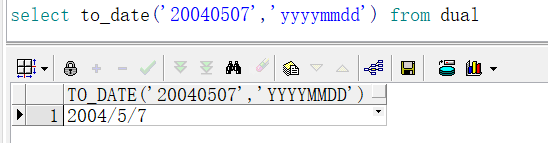
# 八、to\_char()函数

功能：[将日期和数字类型转换成字符类型]



# 九 、to\_date()函数

功能：[将字符类型转换为日期类型]



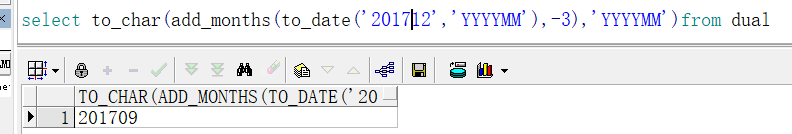
# 十、add\_months（）函数

功能：可以求几个月前或几个月后的时间

格式；add\_months(‘当前时间’，‘月数n’)

月数：为正数时：返回的时当前时间n个月后的时间

为负数时：返回当前时间的n个月前的时间



# 十一、months\_between（）函数

功能：返回两个日期之间的月份数

格式：months\_between（‘日期1’，‘日期2’）

返回的月份数为：日期1-日期2

# 十二 、REGEXP\_LIKE()

功能：和LIKE 共能相似，可以实现模糊查询。但其使用的是正则表达式，所以功能比LIKE更强大，更灵活。

格式：REGEXP\_LIKE(‘字符串’，‘正则表达式’，[取值范围])

3个参数

第一个是输入的字符串

第二个是正则表达式

第三个是取值范围：（可省略）

i：大小写不敏感；

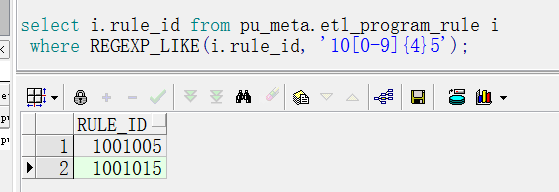
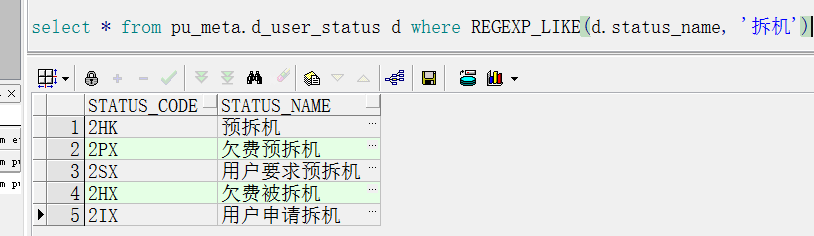
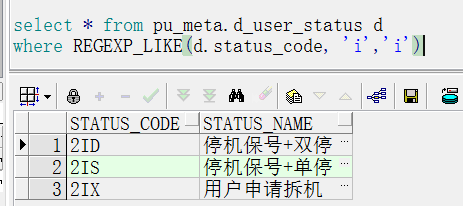
c：大小写敏感；

n：点号 . 不匹配换行符号；

m：多行模式；

x：扩展模式，忽略正则表达式中的空白字符。

|  |
| --- |
| ，REGEXP\_LIKE ：与LIKE的功能相似  2，REGEXP\_INSTR ：与INSTR的功能相似  3，REGEXP\_SUBSTR ：与SUBSTR的功能相似  4，REGEXP\_REPLACE ：与REPLACE的功能相似  它们在用法上与Oracle SQL 函数LIKE、INSTR、SUBSTR 和REPLACE 用法相同，  但是它们使用POSIX 正则表达式代替了老的百分号（%）和通配符（\_）字符。 |

# 十三、（+）的用法

数据库的外连接：

1. 左外连接 (左边的表不加限制)
2. 右外连接(右边的表不加限制)
3. 全外连接(左右两表都不加限制)

外连接(Outer Join)

对应SQL：LEFT/RIGHT/FULL OUTER JOIN。 通常我们省略outer 这个关键字。 写成：LEFT/RIGHT/FULL JOIN。

在左外连接和右外连接时都会以一张表为基表，该表的内容会全部显示，然后加上另外一张表匹配的内容。 如果基表的数据在另一张表没有记录。 那么在相关联的结果集行中列显示为空值（NULL）。

|  |  |  |
| --- | --- | --- |
| 对于外连接， 也可以使用“(+) ”来表示。  1.（+）操作符只能出现在where子句中，并且不能与outer join语法同时使用。    2. 当使用（+）操作符执行外连接时，如果在where子句中包含有多个条件，则必须在所有条件中都包含（+）操作符    3.（+）操作符只适用于列，而不能用在表达式上。    4.（+）操作符不能与or和in操作符一起使用。    5.（+）操作符只能用于实现左外连接和右外连接，而不能用于实现完全外连接。   |  | | --- | | 左连接：select \* from A left join B on A.id=B.id  select \* from A,B where A.id=B.id(+) |  |  | | --- | | 右连接：select \* from A right join B on A.id=B.id  Select \* from A,B where A.id(+)=B.id | |

# 十四、PURGE

SELECT \* FROM RECYCLEBIN; --回收站

FLASHBACK TABLE original\_name TO BEFORE DROP;--恢复

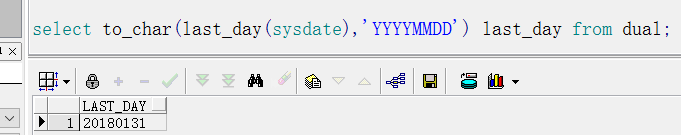
PURGE TABLE original\_name; --从回收站删除

select \* from v$session a where a.SERVICE\_NAME='fxgk'; --进程

# 十五、Last\_day()函数

功能：获取某个日期月份的最后一天时的日期

格式：last\_day(某个日期)



# 十六、row\_number() over()分组排序功能

功能:可是实现分组后，在组内排序，为每一条数据返回一个编号。

格式：row\_number() over( [partition by ‘分组字段’] order by ‘排序字段’)

[partition by ‘分组字段’]：跟具某个字段分组，如果没有指定那么它把整个结果集作为一个分组，它和聚合函数不同的地方在于它能够返回一个分组中的多条记录，而聚合函数一般只有一个反映统计值的记录。 也可以通过该函数实现去重的功能

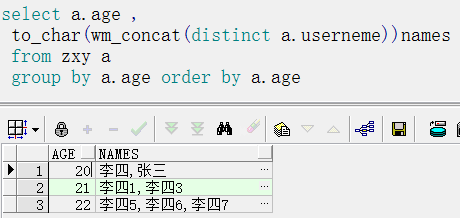
用例：

|  |
| --- |
| Select \*  From (Select A.\*,  ROW\_NUMBER() OVER(Partition By AREA\_CODE1 Order By AMOUNT\_JTHZ Desc) RN  From (  Select T.AREA\_CODE1,  T.ACCT\_ID,  Sum(AMOUNT\_JTHZ) AMOUNT\_JTHZ  From TMP\_OWE\_REC\_ROLLING\_CTRL\_JT2　T  Group By T.AREA\_CODE1, T.ACCT\_ID) A)  Where RN <= 300  选择各地州排名前300 |

# 十七、wm\_concat（）函数

功能：可以将某列的数据合并为一行，数据之间用‘,’隔开

基本格式：wm\_concate(‘字段名’) 配合其它一些函数可以实现复杂功能，可以实现分组排序的功能。



# 十八、Chr()，ascii()

|  |
| --- |
| select ascii('A') from dual; 65 --字符->ascii， select chr('65') from dual; A-- ascii->字符  可以去除数据库中特殊的字符。 |

|  |
| --- |
| 创建视图：  create or replace view ‘视图名’ as  select \* from zxy --查询语句  with read only --只读 |

select nullif(8,1) from dual

select sys\_context('userenv','terminal') from dual; --本机用户名\*/

# 十九、正则表达式

|  |  |
| --- | --- |
| \cx | 匹配由x指明的控制字符。例如， \cM 匹配一个 Control-M 或回车符。x 的值必须为 A-Z 或 a-z 之一。否则，将 c 视为一个原义的 'c' 字符。 |
| \f | 匹配一个换页符。等价于 \x0c 和 \cL。 |
| \n | 匹配一个换行符。等价于 \x0a 和 \cJ。 |
| \r | 匹配一个回车符。等价于 \x0d 和 \cM。 |
| \s | 匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [ \f\n\r\t\v]。 |
| \S | 匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。 |
| \t | 匹配一个制表符。等价于 \x09 和 \cI。 |
| \v | 匹配一个垂直制表符。等价于 \x0b 和 \cK。 |

1. 去除所有的数字以及空格 （将数字替换为空格）

select regexp\_replace(' 1 2 r32 4A2e e ','[0-9]+|\s','') from dual;



1. 只保留数字 （将不是数字的替换为空格）

select regexp\_replace('12 r3 24 A2ee','[^0-9]+|\s','') from dual;

