

# 函数

## case when 搜索

简单case函数

```
select deptno, case deptno
  when 10 then '第一部门'
  when 20 then '第二部门'
  when 30 then '第三部门'
  else '第四部门'
end from emp;
```

case搜索函数

```
select deptno, case
  when deptno=10 then '第一部门'
  when deptno=20 then '第二部门'
  when deptno=30 then '第三部门'
  else '第四部门'
end from emp;
```

case when 生成的列也可以丢到 group by 后面去分组的，where、having都可以使用case表达式需注意的点：

各个分支<表达式>返回的数据类型要统一

case写完后不能丢了end

else可省略但不建议省，没有值时可写 else null

## decode 等值翻译

-- 把10号部门翻译成“十号部门”，20号部门翻译成“二十号部门”，30号部门翻译成“三十号部门”，其他显示其他部门，翻译时可以对列进行处理

```
select decode(deptno, 10, '十号部门', 20, '二十号部门', 30, '三十号部门', '其他部门') from emp
```

decode与case when 的比较：

- 1、decode 只有Oracle、informix才有，其它数据库不支持；
- 2、case when的用法，Oracle、SQL Server、MySQL、informix、都支持；
- 3、decode 只能用做相等判断，但是可以配合sign函数进行大于，小于，等于的判断，CASE

when可用于=,>,<,<=,<>,is null,is not null 等的判断;

4、decode 使用其来比较简洁，CASE 虽然复杂但更为灵活;

5、另外，在decode中，null和null是相等的，但在case when中，只能用is null来判断。

6、decode效率比case when高，Oracle环境中能用decode尽量不用case when

## 行转列

### 聚合case实现行转列

```
-- 求各部门的各职位的工资
-- 各职位 ('SALESMAN'、'MANAGER'、'CLERK') 作为列名
-- 空格后的 salesman、manager、clerk是别名
select deptno,
max(case when job='SALESMAN' then sal end) salesman,
max(case when job='MANAGER' then sal end) manager,
max(case when job='CLERK' then sal end) clerk
from emp group by deptno
```

## pivot 行转列

```
-- 格式
select * from (数据查询集)
    pivot(
        (行转列后列的值) for 需要转的列 in (转换后的列值)
    )
-- 求各部门的各职位的工资
select * from (select deptno,job,sal from emp)
    pivot (
        max(sal) for job in ('SALESMAN','MANAGER','CLERK')
    )
-- 行转列后的值应是聚集函数集
```

unpivot 列转行

## 开窗函数（分析函数）

开窗函数只能写在select里，不能用在where里

### 排序类开窗函数

主要做排序，它能够：

- 在不减少行数的情况下进行计算

- 基于与当前行相关的行集（窗口）进行计算
- 为每行返回一个值，同时保持原始表的行数不变

-- 格式

```
函数名(列名)over(
    partition by 列名
    order by 列名
)
```

其中函数名有：

**row\_number:** 对相等的值不进行区分，相等的值对应的排名不同，序号从1到n连续（其实就是行号）

-- 如果有两个最高分，他们排名序号仍为1和2，而不是并列第一（1,2,3...）

```
select sal,row_number()over(order by sal desc) from emp;
```

**rank:** 相等的值排名相同，但若有相等的值，则序号从1到n不连续

-- 如果有两个最高分，他们序号都为1（并列第一），但下一名的序号仍为3，而不是2

（1,1,3,4...）

```
select sal,rank()over(order by sal desc) from emp
```

**dense:** 对相等的值排名相同，但序号从1到n连续

-- 如果有两个最高分，则下一名的序号为2（即第三个人为第二名）（1,1,2,3）

```
select sal,dense()over(order by sal desc) from emp
```

**dense\_rank:** 密集的排名，紧密排序，可以用来做排序，它序号连续，考虑重复数据，如果值相等序号就相同

-- 第一名有两个最高分，下一名也有两个，则序号为1, 1, 2, 2

```
select sal,dense_rank()over(order by sal desc) from emp
```

**ntile(n):** 将数据分成n组

-- **wm\_concat(列名):** 该函数可以把列值以","号分隔起来,并显示成一行

```
select wm_concat(ename) from emp;
```

--作为普通函数，对工资进行排序，用逗号进行拼接。

```
select listagg(ename,',')within group(order by sal)name from emp;
```

--作为分组函数：

```
select deptno,listagg(ename,',')within group(order by sal)name from emp group
by deptno;
```

--作为分析函数：

```
select deptno,ename,sal,listagg(ename,',')within group(order by
sal)over(partition by deptno)name from emp;
```

## 偏移开窗

```

-- field:需要查找的字段; num:往后查找的行数; defaultvalue:没有符合条件的默认值。
lead(field,num,defaultvalue)

-- lead(params,m,n) 以params为目标向下m位取数，当取不到时默认为n，
select sal,lead(sal,1,0)over(order by sal) lastsal from emp;

-- lag(params,m,n) 以params为目标向上m位取数，当取不到时默认为n
select sal,lag(sal,1,0)over(order by sal) lastsal from emp;

-- 环比 (sal-lastsal/sal *100%
select ename,sal,lag(sal,1,0)over(order by sal) lastsal, concat(
to_char(round((sal-lag(sal,1,0)over(order by sal))/sal*100,2),'990.99') ,'%')
from emp;

```

## 作业练习

```

-- 上机练习8

-- 1.把10号部门翻译成“十号部门”，20号部门翻译成“二十号部门”，30号部门翻译成“三十号部门”
select deptno, case deptno
      when 10 then '十号部门'
      when 20 then '二十号部门'
      when 30 then '三十号部门'
      else '其他部门'
end 部门 from emp

-- 2.给所有的10号部门员工加薪10% 20号部门的员工加薪20% 30号员工加薪30% ,其他部门加薪5%
select ename,deptno,sal,case
      when deptno=10 then sal*1.1
      when deptno=20 then sal*1.2
      when deptno=30 then sal*1.3
      else sal*1.05
end new_sal from emp

-- 3.统计工资级别相应的数量（1600以下 C级,1600-3000 B级,3000以上 A级）
select ename,sal,case
      when sal<1600 then 'C级'
      when sal between 1600 and 3000 then 'B级'
      when sal>3000 then 'A级'
      else null
end 工资级别 from emp

-- 4.实现两种方法的行转列
-- 聚合case

```

```

select deptno,
       max(case when job='SALESMAN' then sal end) salesman,
       max(case when job='MANAGER' then sal end) manager,
       max(case when job='CLERK' then sal end) clerk
from emp group by deptno

-- pivot
select * from (select deptno,job,sal from emp)
              pivot (
                    max(sal) for job in ('SALESMAN','MANAGER','CLERK')
                  )

-- 上机练习9

--1.按照部门编号升序查找所有部门名称，用、隔开
select listagg(dname,', ') within group(order by deptno) from dept

--2.按照工资降序查找每个部门的员工姓名，用、隔开
select deptno,listagg(ename,', ') within group(order by sal desc)
from emp group by deptno order by deptno

--3.使用工资偏移计算环比 (sal-lastsal)/sal*100%
select ename,sal,lag(sal,1,5000)over(order by sal) lastsal,concat(
       to_char(round((sal-lag(sal,1,5000)over(order by
sal))/lag(sal,1,5000)over(order by sal)*100,2),'990.99') , '%')
) from emp

--4.查询员工表中工资最高的前三名
select ename, sal from(
       select ename, sal from emp order by sal desc
     )
where rownum <= 3;

--5.查询员工表中每个部门的工资第2~3名的员工信息
select * from(
       select emp.*,dense_rank()over(partition by deptno order by sal desc)
       rnk from emp
     )
where rnk between 2 and 3

--6.查询员工姓名、部门及部门平均工资,以及部门内最高工资
select ename,deptno,sal,
       round(avg(sal)over(partition by deptno)) 部门平均工资,
       max(sal)over(partition by deptno) 部门最高工资
from emp

```

--7. 每种工作累计求工资和

```
select job,sal from emp order by job,sal
```

```
select job,sum(sal) from emp group by job order by job
```

```
select job,sum(sal)over(partition by job order by sal,ename) from emp
```