

MySQL 中的 DECODE 函数的实现

作者: [PeterChu](#)

原文链接: <https://ld246.com/article/1599242701625>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



在 Oracle 中的 decode() 函数

语法如下：

DECODE (expr, search1, result1[, search2, result2...][, default])

它用于比较参数expr的值，如果匹配到哪一个search条件，就返回对应的result结果，可以有多组search和result的对应关系，如果任何一个search条件都没有匹配到，则返回最后default的值。default参数是可选的，如果没有提供default参数值，当没有匹配到时，将返回NULL。

例：

查询职员表，根据职员的职位计算奖励金额，当职位分别是' MANAGER'、' ANALYST'、' SALSMAN' 时，奖励金额分别是薪水的1.2倍、1.1倍、1.05倍，如果不是这三个职位，则奖励金额取薪值：

```
SELECT ename, job, sal,
       DECODE(job, 'MANAGER', sal * 1.2,
              'ANALYST', sal * 1.1,
              'SALESMAN', sal * 1.05,
              sal
              ) bonus
FROM emp;
```

和DECODE函数功能相似的有CASE语句，实现类似于if-else的操作。

```
SELECT
  ename,
  job,
  sal,
  CASE
    job
```

```

WHEN 'MANAGER' THEN
sal * 1.2
WHEN 'ANALYST' THEN
sal * 1.1
WHEN 'SALESMAN' THEN
sal * 1.05 ELSE sal
END bonus
FROM
emp;

```

DECODE函数在分组查询中的应用

DECODE函数可以按字段内容分组，例如：计算职位的人数，analyst/manager职位属于vip，其余普通员工operation,这种功能无法用GROUP BY简单实现。用decode的实现方式：

```

SELECT DECODE(job, 'ANALYST', 'VIP',
              'MANAGER', 'VIP',
              'OPERATION') job,
COUNT(1) job_cnt
FROM emp
GROUP BY DECODE(job, 'ANALYST', 'VIP', 'MANAGER', 'VIP', 'OPERATION');

```

DECODE函数也可以按字段内容排序，例如：Dept表中按“研发部”、“市场部”、“销售部”排，用普通的select语句，无法按照字面数据排序：

```

SELECT deptno, dname, loc
FROM dept
ORDER BY
  DECODE(dname, '研发部',1,'市场部',2,'销售部',3), loc;

```

MySQL 中实现

在 MySQL 中可以使用 `if()` 函数完成需要使用 DECODE() 函数的简单情景，复杂一点的可以使用 CAS () 函数。

case函数和代码里面 if...else...很是相似，个人感觉这就是一个判断选择函数。

语法：

`case...when...then...else...end`

在这个语法基础上面有书写两种格式：简单case函数和case搜索函数。

另外有一点，case函数和 if 函数一样，只要一个判断满足了，后面剩下的case部分将会被自动忽略不再匹配。所以写表达式的时候要特别注意一下。

-- 计算职位的人数，其中，“ANALYST”和“MANAGER”职位用“VIP”表示，其余是普通员工，位用“OPERATION”显示。分别计算VIP职位和普通职员职位的人数

```

SELECT * FROM emp ORDER BY job;
SELECT empno, ename, IF((job IN ('ANALYST','MANAGER')), 'VIP', 'OPERATION') JOB FROM e
mp;
SELECT IF((job IN ('ANALYST','MANAGER')), 'VIP', 'OPERATION') JOB, COUNT(1) JOB_COUNT

```

```

FROM emp GROUP BY IF((job IN ('ANALYST','MANAGER')), 'VIP', 'OPERATION');
SELECT CASE job WHEN 'ANALYST' THEN 'VIP' WHEN 'MANAGER' THEN 'VIP' ELSE 'OPERATION' END AS JOB, COUNT(1) JOB_COUNT FROM emp GROUP BY CASE job WHEN 'ANALYST' THEN 'VIP' WHEN 'MANAGER' THEN 'VIP' ELSE 'OPERATION' END;
SELECT empno, ename, CASE job WHEN 'ANALYST' THEN 'VIP' WHEN 'MANAGER' THEN 'VIP' ELSE 'OPERATION' END AS JOB FROM emp;
SELECT empno, ename, CASE WHEN job IN ('ANALYST','MANAGER') THEN 'VIP' ELSE 'OPERATION' END AS JOB FROM emp;
SELECT CASE WHEN job IN ('ANALYST', 'MANAGER') THEN 'VIP' ELSE 'OPERATION' END AS JOB, COUNT(*) AS JOB_COUNT FROM emp GROUP BY CASE WHEN job IN ('ANALYST', 'MANAGER') THEN 'VIP' ELSE 'OPERATION' END; -- 不能忘记 END, 否则会报错

```

case函数还可以与一些统计函数连用。

```

SELECT
    SUM(CASE sex WHEN 1 THEN 1 ELSE 0 END ) AS '男生',
    SUM(CASE sex WHEN 0 THEN 1 ELSE 0 END ) AS '女生',
    SUM(CASE WHEN sex=1 OR sex=0 THEN 0 ELSE 1 END) AS '未知'
FROM
    'tb_student';

```

结果:

| 男生 | 女生 | 未知 |
|----|----|----|
| 2 | 2 | 1 |

之前的示例使用 SUM() 统计的 SQL 写法。

```

SELECT
    SUM(CASE WHEN job IN ('ANALYST','MANAGER') THEN 1 ELSE 0 END) AS VIP,
    SUM(CASE WHEN job NOT IN ('ANALYST','MANAGER') THEN 1 ELSE 0 END ) AS OPERATION
FROM emp;

```

case 在排序子句中的使用

```

-- 将dept 表中的数据按 'OPERATIONS' 、 'ACCOUNTING' 、 'SALES' 进行自定义排序
SELECT deptno, dname, loc FROM dept ORDER BY DECODE(dname, 'OPERATIONS', 1, 'ACCOUNTING', 2, 'SALES', 3); -- Oracle 中用法
SELECT deptno, dname, loc FROM dept ORDER BY CASE dname WHEN 'OPERATIONS' THEN 1 WHEN 'ACCOUNTING' THEN 2 WHEN 'SALES' THEN 3 END ;

```

参考: [MySQL函数之CASE](#)

[Mysql——case函数](#)