

# 实用数据库4

2018年3月9日 13:34

## 1. 处理NULL的函数

### a. NVL(expr1, expr2)

如果expr1为空，就显示expr2

```
SQL> select empno, ename, nvl(comm, 0) from scott.emp;
```

EMPNO	ENAME	NVL (COMM, 0)
7369	SMITH	0
7499	ALLEN	300
7521	WARD	500
7566	JONES	0
7654	MARTIN	1400
7698	BLAKE	0
7782	CLARK	0
7788	SCOTT	0
7839	KING	0

要显示与该列不同类型的内容:

```
SQL> select empno, ename, nvl(comm, 0), nvl(to_char(mgr), 'boss') from scott.emp;
```

EMPNO	ENAME	NVL (COMM, 0)	NVL (TO_CHAR (MGR), ' BOSS' )
7369	SMITH	0	7902
7499	ALLEN	300	7698
7521	WARD	500	7698
7566	JONES	0	7839
7654	MARTIN	1400	7698
7698	BLAKE	0	7839
7782	CLARK	0	7839
7788	SCOTT	0	7566
7839	KING	0	boss
7844	TURNER	0	7698
7876	ADAMS	0	7788
7900	JAMES	0	7698
7902	FORD	0	7566
7934	MILLER	0	7782

### b. nvl2(expr1, expr2, expr3):

expr1为空，显示expr2，否则显示expr3

```
SQL> select ename, sal, comm, nvl2(comm, sal + comm, sal) from scott.emp;
```

ENAME	SAL	COMM	NVL2 (COMM, SAL+COMM, SAL)
SMITH	2800		2800
ALLEN	1600	300	1900
WARD	1250	500	1750
JONES	2975		2975
MARTIN	1250	1400	2650
BLAKE	2850		2850
CLARK	2450		2450
SCOTT	3000		3000
KING	5000		5000
TURNER	1500	0	1500
ADAMS	1100		1100

- c. Nullif()
- d. coalesce(expr1, expr2, expr3, ..., exprn):  
expr1为空, 显示expr2, 依此类推

```
SQL> select ename, sal, comm, coalesce(comm + sal, sal, comm) income from scott.emp;
```

ENAME	SAL	COMM	INCOME
SMITH	2800		2800
ALLEN	1600	300	1900
WARD	1250	500	1750
JONES	2975		2975
MARTIN	1250	1400	2650
BLAKE	2850		2850
CLARK	2450		2450
SCOTT	3000		3000
KING	5000		5000
TURNER	1500	0	1500
ADAMS	1100		1100

  

ENAME	SAL	COMM	INCOME
JAMES	950		950
FORD	3000		3000
MILLER	1300		1300

2. 获取年份: to\_number(to\_char(sysdate,'yyyy'))

3. 实现IF-THEN-ELSE逻辑的两种方法:

- a. CASE语句
- b. **DECODE()** 函数

```
SELECT last name, job id, salary,
       DECODE(job_id, 'IT_PROG', 1.10*salary,
                  'ST_CLERK', 1.15*salary,
                  'SA_REP', 1.20*salary,
                  salary)
       REVISED_SALARY
FROM employees;
```

```
SQL> select count(*) TOTAL,
2      sum(decode(to_char(hire_date, 'yyyy'), '2001', 1, 0)) "2001",
3      sum(decode(to_char(hire_date, 'yyyy'), '2002', 1, 0)) "2002",
4      sum(decode(to_char(hire_date, 'yyyy'), '2003', 1, 0)) "2003"
5  from hr.employees;
```

TOTAL	2001	2002	2003
107	1	7	6

#### 4. 组函数:

- WHERE —— 记录筛选
- HAVING —— 分组筛选

#### 5. 最大&最小

- greatest()

```
SQL> select greatest(1, 4, 7) MAX from dual;
```

MAX
7

- least()

```
SQL> select least(1, 4, 7) MIN from dual;
```

MIN
1

#### 6. 对分组进行小计

- Rollup

```
SQL> select department_id dept_id, job_id, sum(salary)
2  from hr.employees
3  group by rollup(department_id, job_id)
4  order by 1;
```

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
10		4400
20	MK_MAN	13000
20	MK_REP	6000
20		19000
30	PU_CLERK	13900
30	PU_MAN	11000
30		24900
40	HR_REP	6500
40		6500
50	SH_CLERK	64300
50	ST_CLERK	55700
50	ST_MAN	36400
50		156400
60	IT_PROG	28800
60		28800
70	PR_REP	10000
70		10000

b. Cube()

```
SQL> select department_id dept_id, job_id, sum(salary)
2  from employees
3  group by cube(department_id, job_id)
4  order by 1;
```

DEPT_ID	JOB_ID	SUM(SALARY)
	AD_PREP	24000
	AD_VP	34000
	FI_ACCOUNT	39600
	FI_MGR	12008
	HR_REP	6500
	IT_PROG	28800
	MK_MAN	13000
	MK_REP	6000
	PR_REP	10000
	PU_CLERK	13900
	PU_MAN	11000
	SA_MAN	61000
	SA_REP	7000
	SA_REP	250500
	SH_CLERK	64300
	ST_CLERK	55700
	ST_MAN	36400
		7000
		691416

7. Sqlldr: Oracle加载文件的工具 —— 在操作系统下(terminal)做

```

C:\WINDOWS\system32>sqlldr hr/hr control='C:\Users\Johnson Chen\Desktop\st.ctl'
SQL*Loader: Release 11.2.0.1.0 - Production on 星期五 3月 9 14:51:43 2018
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
达到提交点 - 逻辑记录计数 7

C:\WINDOWS\system32>sqlplus hr/hr
SQL*Plus: Release 11.2.0.1.0 Production on Fri Mar 9 14:51:50 2018
Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> select * from st;

```

NAME	SUBJECT	SCORE
frant	chinese	100
frant	english	99
frant	math	98
castle	chinese	97
castle	english	96
castle	math	95

### st.csv 文件

```

name,subject,score
frant,chinese,100
frant,english,99
frant,math,98
castle,chinese,97
castle,english,96
castle,math,95

```

### st.ctl 文件

```

load
infile 'C:\Users\Johnson Chen\Desktop\st.csv'
into table hr.st
(name char terminated by ',',
subject char terminated by ',',
score integer external terminated by ',')

select name,
       sum(decode(subject, 'chinese', score, 0)) CHINESE,
       sum(decode(subject, 'math', score, 0)) MATH,

```

```

sum(decode(subject, 'english', score, 0)) ENGLISH
from st
group by name;
SQL> select name,
2      sum(decode(subject, 'chinese', score, 0)) CHINESE,
3      sum(decode(subject, 'math', score, 0)) MATH,
4      sum(decode(subject, 'english', score, 0)) ENGLISH
5  from st
6  group by name;

```

NAME	CHINESE	MATH	ENGLISH
castle	97	95	96
frant	100	98	99

## 8. 尽量减少访问表的次数：把小表写在后边！

Nested loop

dept(10, 20, 30, 40)

emp(10w)

- From emp, dept

先去 dept 找部门编号为10的员工，然后再去emp中找对应的。总共查表4次

- From dept, emp —— 查找10w次

## 9. 表连接

能够用内连接实现的，不要用外连接，因为外连接开销远大于内连接

左外连接

```

select b.buyer_id, b.buyer_name, s.qty
from buyers b, sales s
where b.buyer_id(+) = s.buyer_id;

```

全外链接

```

select b.buyer_id, b.buyer_name, s.qty
From buyers b full outer join sales s
on b.buyer_id = s.buyer_id;

```

## 10.

```

Select a.buyer_id as buyer1, a.prod_id, b.buyer_id as buyer2
From sales a, sales b
Where a.prod_id = b.prod_id
And a.buyer_id <> b.buyer_id;

```

交叉连接

- Select b.buyer\_name, s.qty from buyers b, sales s;
- Select b.buyer\_name, s.qty from buyers b cross join sales s;

#### 11. 更改列宽

```
col 职工姓名 format a30
```

**P182**