<https://blog.csdn.net/sdut406/article/details/82795585> (DECODE)

查询当前的连接Oracle

select \* from (select SID,AUDSID,USER#,USERNAME,STATUS,SCHEMANAME,

OSUSER,MACHINE,TERMINAL,PROGRAM,MODULE,LOGON\_TIME,PREV\_EXEC\_START as "last\_time",

BLOCKING\_SESSION\_STATUS as "bss",SERVICE\_NAME,SQL\_ID

from v$session

where audsid not in (0)

and program !='plsqldev.exe'

and osuser !='Administrator'

order by LOGON\_TIME desc)

group by machine

# oracle 常用的sql 使用

## DECODE

表达的含义：

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

### 导入一张表

测试数据表格

-- ----------------------------

-- Table structure for STUDENT2

-- ----------------------------

DROP TABLE "STUDENT2";

CREATE TABLE "STUDENT2" (

"ID" NUMBER(4) NOT NULL ,

"NAME" VARCHAR2(10 BYTE) NULL ,

"AGE" NUMBER(4) NULL ,

"SEX" NUMBER(2) NULL

)

LOGGING

NOCOMPRESS

NOCACHE

;

-- ----------------------------

-- Records of STUDENT2

-- ----------------------------

INSERT INTO "STUDENT2" VALUES ('1', '张三', '15', '1');

INSERT INTO "STUDENT2" VALUES ('2', '李四', '20', '2');

INSERT INTO "STUDENT2" VALUES ('3', '王儿', '19', '2');

INSERT INTO "STUDENT2" VALUES ('4', '赵六', '30', '1');

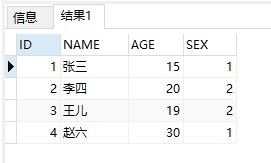
-- ----------------------------

-- Checks structure for table STUDENT2

-- ----------------------------

ALTER TABLE "STUDENT2" ADD CHECK ("ID" IS NOT NULL);

表中的数据



查询出的数据，1表示男生，2表示女生

SELECT T.ID, T.NAME, T.AGE, DECODE(T.SEX,

'1', '男',

'2', '女',

'其他') SEX

FROM STUDENT2 T;

等价于 case when then end

SELECT T.ID,

T.NAME,

T.AGE,

CASE T.SEX

WHEN 1 THEN

'男'

WHEN 2 THEN

'女'

ELSE

'其他'

END SEX

FROM STUDENT2 T;

这种方式，同样适用于 mysql

## 比较大小

说明：sign(value)函数会根据value的值为0，正数，负数，分别返回0，1,-1

数据：

举例：

select sign(20) from dual; *-- 1*

select sign(0) from dual; *-- 0*

select sign(-1) from dual; *-- -1*

需求：年龄在20以上的显示20以上，20以下的显示20以下，20的显示正好20

SELECT

T . ID,

T . NAME,

DECODE (

SIGN (T.age-20),

1,'大于20',

0, '等于 20',

'小于20'

) age,

T .sex

FROM

student2 T;

## instr 字符串查找

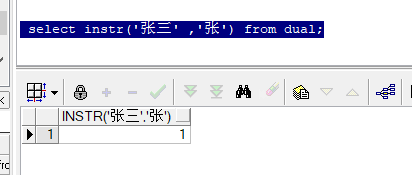
格式一：instr( string1, string2 ) / instr(源字符串, 目标字符串)

格式二：instr( string1, string2 [, start\_position [, nth\_appearance ] ] ) / instr(源字符串, 目标字符串, 起始位置, 匹配序号

返回的结果是： 目标字符所在的位置

0 表示不包含

1 表示 位置是 1



Demo

-- 查询是否 为张三

SELECT T.ID, T.NAME,

DECODE(INSTR(T.NAME, '张三'),

0, '否',

'是') AS HOST,

T.AGE, T.SEX

FROM STUDENT2 T;

## concat 字符串拼接

格式：instr( string1, string2 )

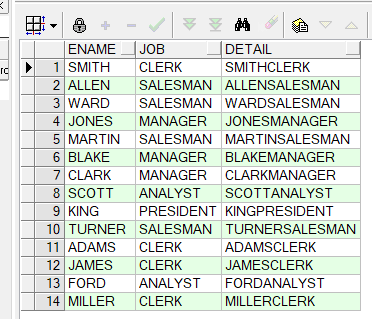
只能拼接两个字符串

SELECT T.ENAME, T.JOB, CONCAT(T.ENAME, T.JOB) AS DETAIL FROM EMP T;

## 多字符串 使用 || 进行拼接

SELECT T.ENAME, T.JOB, T.ENAME || T.JOB AS DETAIL FROM EMP T;

运的结果的 是



也可以进拼接多个 字符串

SELECT T.ENAME, T.JOB, T.ENAME || ':' || T.JOB AS DETAIL FROM EMP T;

## NVL nvl2 nullif coalesce 空判断

select nvl(7, 6) from dual; *---第一个参数为null显示第二个，否则显示第一个*

select nvl2(3, 1, 8) from dual; *---第一个参数为null显示第三个，否则显示第二个*

select (8, 8) from dual; *---两个参数一样显示null。不一样显示第一个*

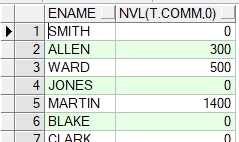
select coalesce(null, 1, null, 2, null, 3) from dual;*---输出第一个不为null的值*

NVL(eExpression1, eExpression2)

判断 eExpression1 为空 返回 eExpression2

-- 判断 emp 表 comm 奖金 null 返回 0

SELECT T.ENAME, NVL(T. COMM, 0) FROM EMP T



## 字符串

字符串 的相关的

*------字符函数*

select lower('YES') from dual;*---大写变小写*

select upper('yes') from dual;*---小写变大写*

select substr('abcdefg', 3, 2) from dual;*---截取字符串 从第三个 开始截 取2 个*

select length('abcdefg') from dual;*---获取字符串长度*

*----第二个字符串在第一个字符串中的位置*

select instr('ababcdefg', 'ab') from dual;

*----用第三个字符在左边补齐第一个字符串，长度是第二个参数*

select lpad('abc', 6, 'h') from dual;

*----去空格*

select trim(' a '), ' a ' from dual;

*---替换字符串*

select replace('abcdefg', 'ab', 'sss') from dual;

## 数值函数 四舍五入

----数值函数

select round(26.16, 1) from dual;--四舍五入

select round(11126.16, -2) from dual;--四舍五入

-- 直接截取值

select trunc(26.16, 1) from dual;

-- 求余数

select mod(12, 5) from dual;

## 时间查询

时间相关的函数

---只取年【英文】

select to\_char(sysdate, 'year') from dual;--- twenty eighteen

---只取年

select to\_char(sysdate, 'yyyy') from dual; -- 2019

---只取月

select to\_char(sysdate, 'mm') from dual;---05

---只取月【英文】

select to\_char(sysdate, 'month') from dual;---5月

---周几

select to\_char(sysdate, 'day') from dual;--- 星期四

---只取天

select to\_char(sysdate, 'dd') from dual;---23

---只取周号【简写】

select to\_char(sysdate, 'dy') from dual ;---星期四

-- 日期

select to\_char(sysdate, 'yyyy-mm-dd hh24:mi:ss') from dual; -- 2019-05-23 15:28:31

-- 转成时间

select to\_date('2019-05-23 15:28:31', 'yyyy-mm-dd hh24:mi:ss') from dual; -- 2017/3/4 17:55:27

查询时间差

计算两个时间差

select sysdate , to\_date('2018-05-23 15:28:31', 'yyyy-mm-dd hh24:mi:ss') from dual;

月数

SELECT MONTHS\_BETWEEN(SYSDATE, TO\_DATE('2018-05-23 02:28:31', 'yyyy-mm-dd hh24:mi:ss')) MONTHS\_COUNT

FROM DUAL;

天数 （两个天数， 直接剪发）

SELECT

SYSDATE -

TO\_DATE (

'2018-05-23 15:28:31',

'yyyy-mm-dd hh24:mi:ss'

)

FROM

dual;

运行结果是



天数 整数 （增加一个四舍五入）

SELECT

ROUND (

SYSDATE - TO\_DATE (

'2018-05-23 15:28:31',

'yyyy-mm-dd hh24:mi:ss'

)

)

FROM

dual;

计算 星期数 （计算天数 /7 。 然后再 取整 即可）

SELECT ROUND((SYSDATE - TO\_DATE('2018-05-23 15:28:31', 'yyyy-mm-dd hh24:mi:ss')) / 7)

FROM DUAL;

昨天

SELECT SYSDATE - 1 FROM DUAL;

查询当天

<https://blog.csdn.net/huangbaokang/article/details/80564530>

sysdate+1 加一天

sysdate+1/24 加1小时

sysdate+1/(24\*60) 加1分钟

sysdate+1/(24\*60\*60) 加1秒钟

sysdate-1 减一天

sysdate-1/24 减1小时

sysdate-1/(24\*60)减1分钟

sysdate-1/(24\*60\*60) 减1 秒

SELECT (sysdate-10/1440) from dual;

* 查询十分钟前

---------------------

作者：rjx贤

来源：CSDN

原文：https://blog.csdn.net/baidu\_41660182/article/details/84970079

版权声明：本文为博主原创文章，转载请附上博文链接！

## 聚合函数的使用

常用聚合函数

select count(1) from emp e; *-- 记录的条数*

select sum(e.sal) from emp e; *-- 工资总数*

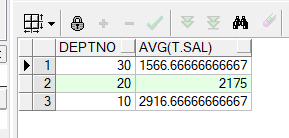
select max(e.sal) from emp e; *-- 最大值*

select min(e.sal) from emp e; *-- 最小值*

select avg(e.sal) from emp e; *-- 平均值*

查询 每个部门的平均工资

SELECT T.DEPTNO, AVG(T.SAL) FROM EMP T GROUP BY T.DEPTNO;



增加条件 部门工资 大于 200 元

SELECT T.DEPTNO, AVG(T.SAL) FROM EMP T GROUP BY T.DEPTNO HAVING AVG(T.SAL) > 2000;

聚合函数 ，增加条件用 having

查询大于 1000 元，部门平均工资

SELECT T.DEPTNO, AVG(T.SAL) FROM EMP T

WHERE T.SAL > 1000

GROUP BY T.DEPTNO

## in not in 用法

in 表示 在记录内

not in 记录中没有

注意： in ，not in 后面跟的是集合，集合后面不能有 null 的值

-总结：用in或者not in的时候，后面集合的条件必须加 is not null

-- 查询所有领导

select mgr from emp where mgr is not null;

-- 排除所有的领导 记录

SELECT \* FROM EMP WHERE EMPNO NOT IN

(SELECT MGR FROM EMP WHERE MGR IS NOT NULL)

## exits

exits(查询语句)

查询语句：不为null -- 执行前面的

结果为null 不执行结果

简单来说 exits( 查询语句)

查询语句 ，有结果 相当于 1 = 1

没有查询到结果 相当于 1 = 2

用法一：判断exists后面的sql语句是否为真

sql查询语句为空 返回值是false

sql查询语句有值 返回值就是true

select \* from emp where exists (select \* from dept where deptno=1)

select \* from emp where exists (select \* from dept where deptno=10)

第一条sql 查询结果是 空

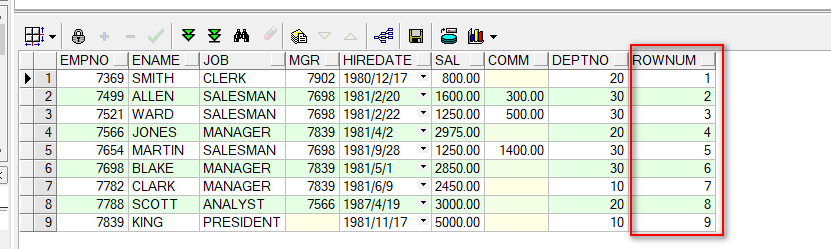
第二条 查询结果是： select \* from emp

## rownum 伪列：

在执行select操作的时候，每加载一行数据

--------------------就在该数据上加上一个序列

SELECT T.\* , rownum FROM emp T



从上面可以看出 rownum 是一列 从 0 开始数字

**分页**

可以利用rownum 进行分页

比如查询 rownum 5-10 数据

比如：

SELECT T.\*, ROWNUM

FROM EMP T

WHERE ROWNUM > 5

运行结果是空 ，rownum 只能是小于

如何实现分页，

1. 先查询前 10 行
2. 通过子查询， 查询 大于 5 行内容

-- 先查询前 6 行

SELECT T.\*, ROWNUM rn

FROM EMP T where ROWNUM < 6

-- 通过子查询查询

SELECT \*

FROM (SELECT T.\*, ROWNUM RN FROM EMP T WHERE ROWNUM < 6)

WHERE RN >= 3

这样就可以达到分页的效果

## 集合的使用

-------union all 联合所有

----union 联合去重

-------intersect 交集

-----minus 差集

部门 10 和 20 员工

SELECT T.\*,T.ROWID FROM emp T where t.deptno = 20

union all

SELECT T.\*,T.ROWID FROM emp T where t.deptno > 10;

这样 部门20 出现 2 遍

SELECT T.\*,T.ROWID FROM emp T where t.deptno = 20

union *-- all*

SELECT T.\*,T.ROWID FROM emp T where t.deptno > 10;

这样 去掉上面重复的

## 历史记录（查询某个时间点的记录）

资料来源： <https://www.cnblogs.com/moonandstar08/p/6146935.html>

select \* from 表名称 as of timestamp to\_timestamp('2019-08-01 15:01:00','yyyy-mm-dd hh24:mi:ss')

## 恢复记录（查询某个时间点的记录）

delete from t\_kft\_bank;

insert into t\_kft\_bank

select \* from t\_kft\_bank as of timestamp to\_timestamp('20191112 11:06:00', 'yyyymmdd HH:mi:ss') commit;

资料来源： 版权声明：本文为CSDN博主「风中浮柳」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/fefsdf/article/details/53027204>