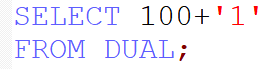
# 运算符

## 1.算术预算符

**+ - \* (/ div) (% mode)**

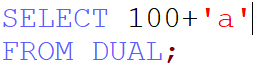
**运用在select语句的字段中、where语句中**

**在sql中， + 没有连接的作用，就表示为加法运算。此时，会将字符串转化为数值（隐式转换）**

****

****

**然而像**

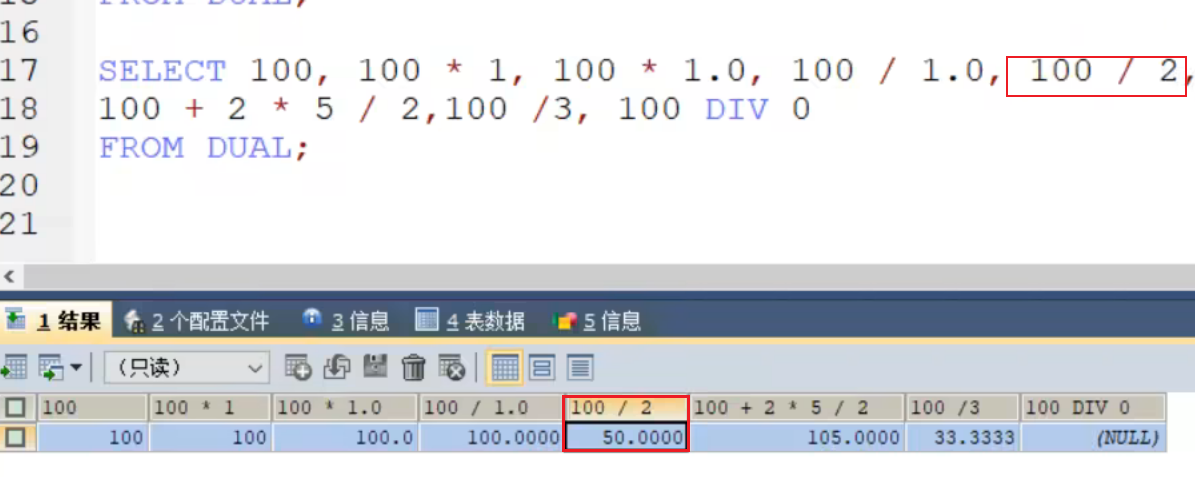
****

**这种，sql会先尝试将其转化为数值，由于字符’a’并没有对应所谓的数值，转换失败，**

****

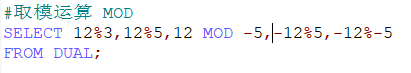
**因此直接将a转化为0**

### /除法中

****

**sql会默认除不尽，自动给你保存四位小数**

### % MOD 模

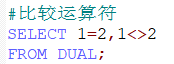




我们发现，模的结果符号与被模数的正负相同。

## 2.比较运算符

比较运算符用来对表达式左边的操作数和右边的操作数进行比较，比较的结果为真则返回1，比较的结果 为假则返回0，其他情况则返回NULL。





### =

数值类型的进行比较：结果为真返回1，结果为假返回0





既然这里1=’a’结果不为null。说明’a’进行了隐式转换，那转换为了多少呢？



明显，若字符串不能转换为数值时，会隐式转换为0

那字符串和字符串进行比较呢？



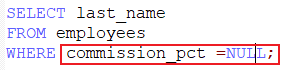


明显，比较运算符两端都为字符或字符串时，并不会进行隐式转换。其比较的是每个字符串中字符的ASCII码值是否相等。



null参与都是null

那我能不能

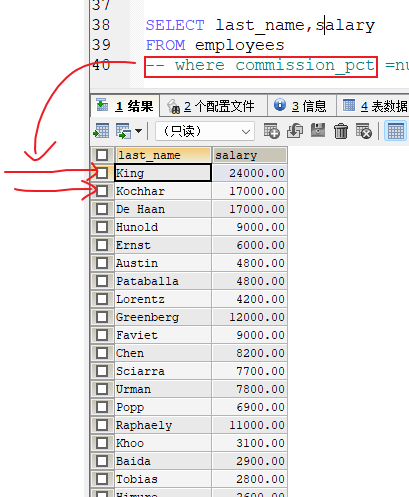


查询到值为null的记录呢？



否定。

实际上sql的查询策略是



指针一条一条的和where的条件进行比对，若结果为真则呈现出来，若不为则都将过滤掉

那么，我们知道 commission\_pct中有很多null，当null=null运算式出现时，返回的是null，不满足为真，自然也都被过滤掉了。

WHERE判断为真时才会呈现记录，否则不呈现！

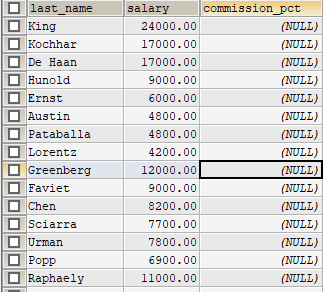
### <=> 安全运算符

正是为了解决null参与运算的问题！其他和=并没有区别

当两个操作数为null时，返回1

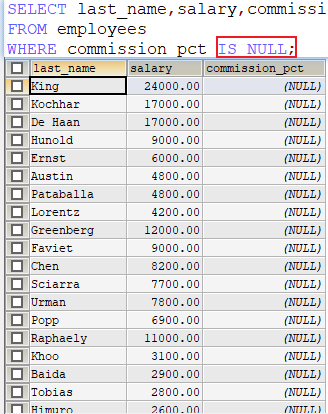
当一个操作数为null时，返回0



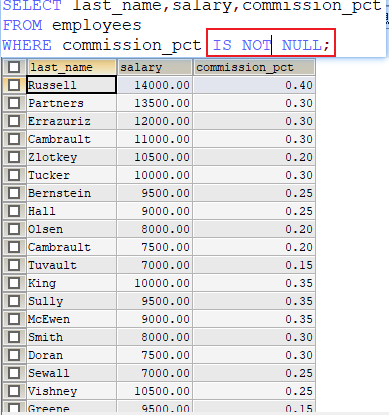


### 非符号类运算符

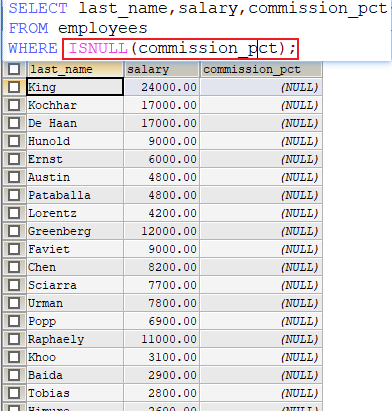




此等价于 <=> null



查找不为null的记录



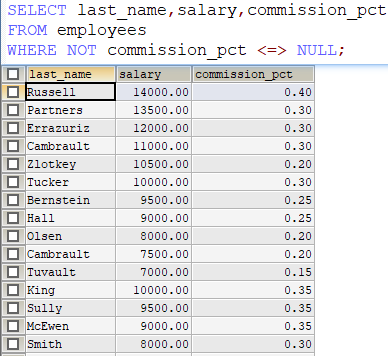
isnull()函数，若括号内的值为null则函数返回1

那在



这种基础上，有办法找到值不为null的记录吗？

注意：并没有安全不等于 !<=>



where not 表达式

就能将后面的条件反转

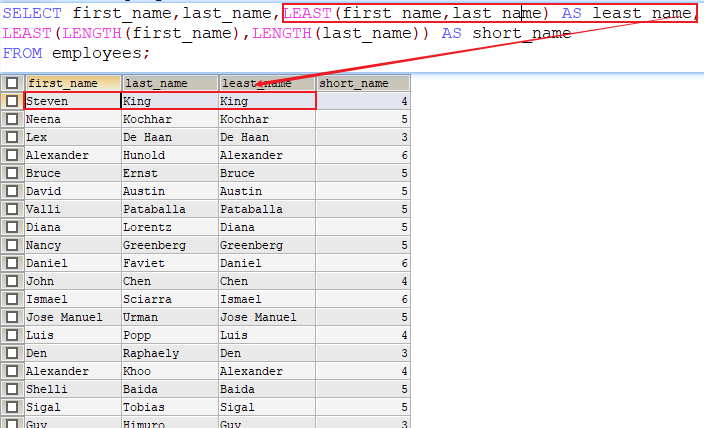
### LEAST()/GREATEST()





同为字符比较的是ASCII码值

里面可放多个数值

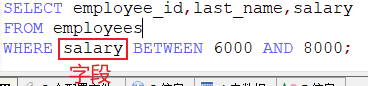


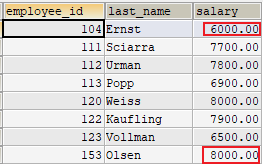
least内输入字段，字符串的比较规则和C相同

这是自己的属性进行比较，并不是不同人之间的相同属性进行比较

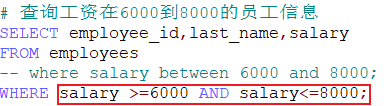
### BETWEEN AND

查询区间运算符

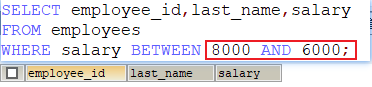




左右区间是包含着得了。



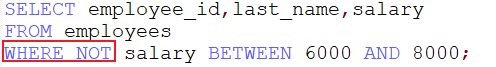
当然这种方式也是可以的



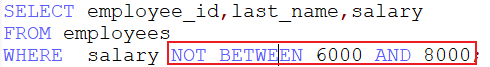
当然反转后就不行了，

between …… and…… 前一个是上界，后一个是下界

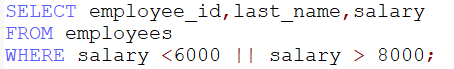
若要找不再6000到8000内的



或者



或者

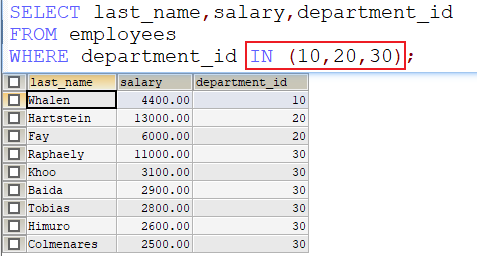


or ||

### IN(set)/NOT IN(set)

对于离散值的查找

set：集合、集

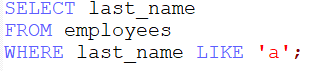


集合内的各个数是或的关系，满足其一便为真

not in()则相反，满足集合内的一条则为假

### LIKE

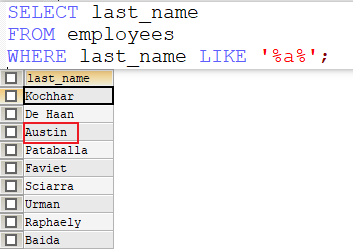
模糊查询



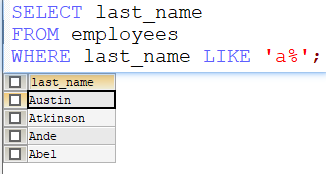
如果是只是单纯加个LIKE的话，那效果和=是一样的。

下面这些符号有占据位置的意思（我自己的理解）

% 代表不确定个数的字符（包含0,1,2，多个都是可以的），需用两个百分号（% %）表示

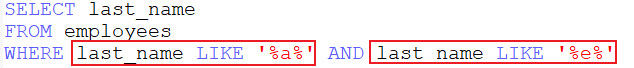


若是找开头为a的，则应LIKE ‘a%’



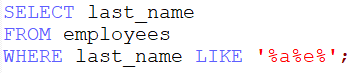
查询同时包含a和e呢

土方法



我们可以直接在%a%后面再补一个e

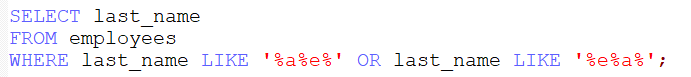
%a%e%



但是这样查询的

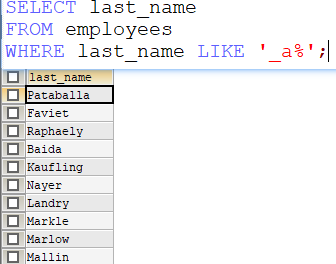


a会在e前面，因此要包括另一种情况



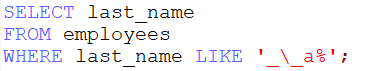
\_ 代表一个不确定的字符

查询第二个字符是'a'的员工信息



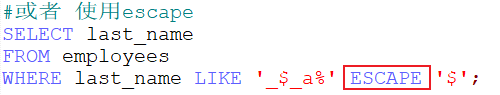
那若名字里包含下划线怎么办呢？

当然是使用转义字符\了！





另一种写法：



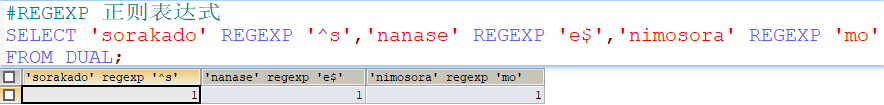
你想用其他当转义字符的话，需要用escape定义这个转义字符；若用的\，则要省略escape

### REGEXP

regular expression：正则表达式

（1）‘^’匹配以该字符后面的字符开头的字符串。

（2）‘$’匹配以该字符前面的字符结尾的字符串。



再次强调：regexp为运算符关键字，整句构成一个表达式，若判断为真返回1，若判断为假返回0，因此此句可看成

SELECT 1,1,1 FROM DUAL;

而select后遇到常量则会填充满字段记录