

[数据库] SQL查询语句表行列转换及一行数据转换成两列

原创 Eastmount 2016-01-22 05:03:53 26261 收藏 1

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、



Eastmount

¥9.90

订阅

本文主要讲述了SQL查询语句表之间的行列转换，同时也包括如何将一行数据转换成两列数据的方法、子查询的应用、decode函数的用法。希望文章对你有所帮助~

- 1.创建数据库表及插入数据
- 2.子查询统计不同性质的学生总数
- 3.一行数据转换成两列数据 union all
- 4.表行列数据转换(表转置)

1.创建数据库表及插入数据

创建数据库、创建学生表并设置主键、插入数据代码如下：

```
-- 创建数据库
create database StudentMS

-- 使用数据库
use StudentMS

-- 创建学生表 (属性: 姓名、学号(pk)、学院、出生日期、性别、籍贯)
create table xs
(
    name varchar(10) not null,
    id varchar(10) not null,
    xy varchar(10),
    birthday datetime,
    xb char(2),
    jg varchar(8)
)

-- 创建学生表主键: 学号
alter table xs
add constraint
pk_xs primary key(id)

-- 插入数据
insert into xs
(id, name, xb, birthday, xy, jg)
values('1160001', '刘备', '男', '1991-11-5', '软件学院', '河北省');
```

输出数据如下图所示：

	name	id	xy	birthday	xb	jg
1	刘备	1160001	软件学院	1991-11-05 00:00:00.000	男	河北省
2	马超	1160002	软件学院	1992-01-17 00:00:00.000	男	河南省
3	黄月英	1160003	软件学院	1991-04-17 00:00:00.000	女	湖北省
4	关羽	1160004	计算机学院	NULL	男	山西省
5	孙尚香	1160005	计算机学院	1993-07-07 00:00:00.000	女	江苏省
6	曹操	1160006	计算机学院	1991-10-23 00:00:00.000	男	河南省
7	夏侯惇	1160007	计算机学院	NULL	男	NULL
8	孙权	1160008	自动化学院	1991-12-01 00:00:00.000	男	浙江省
9	甄姬	1160009	自动化学院	1991-12-01 00:00:00.000	女	贵州省
10	黄盖	1160010	自动化学院	1992-10-19 00:00:00.000	男	浙江省

2.子查询统计不同性质的学生总数

使用子查询统计不同学院总人数、不同性别总人数和河北/河南学生总人数。

```
-- 子查询统计人数
select a.a_num as 软院人数, b.b_num as 计院人数, c.c_num as 自动化人数,
       d.d_num as 男生人数, e.e_num as 女生人数, f.f_num as 河北河南人数
from
(select count(*) as a_num from xs where xy='软件学院') a,
(select count(*) as b_num from xs where xy='计算机学院') b,
(select count(*) as c_num from xs where xy='自动化学院') c,
(select count(*) as d_num from xs where xb='男') d,
(select count(*) as e_num from xs where xb='女') e,
(select count(*) as f_num from xs where jg in ('河北省','河南省')) f;
```

输出结果：

	软院人数	计院人数	自动化人数	男生人数	女生人数	河北河南人数
1	3	4	3	7	3	3

PS：若中文汉字太长报错，则需引用双引号。如：select num as "项目(文化学术讲座)"

3.一行数据转换成两列数据

这时，项目SQL语句的需要是显示成两列如下图所示：

	统计类别	数量
1	软院人数	3
2	计院人数	4
3	自动化人数	3
4	男生人数	7
5	女生人数	3
6	河北河南人数	3

其实简单编写SQL语句，前端再处理这些数据更加方便，当然SQL也是能处理的。

当时走进了一个误区，认为"软件人数"是select中as自定义的一行数据的属性，如何显示在表中呢？当时是通过Oracle方法decode自定义显示的，其实直接输出，union all取代子查询即可。当然union all其它表也可以继续添加。

```
select '软院人数' as "统计类别", count(*) as "数量" from xs where xy='软件学院'
union all
select '计院人数', count(*) from xs where xy='计算机学院'
union all
select '自动化人数', count(*) from xs where xy='自动化学院'
union all
select '男生人数', count(*) from xs where xb='男'
union all
```

```
select '女生人数', count(*) from xs where xb='女'
union all
select '河北河南人数', count(*) from xs where jg in ('河北省','河南省');
```

这里我简单给大家回顾下UNION ALL方法: (参考: [MIN飞翔博客](#))

UNION:

- (1) 其目的是将两个SQL语句的结果合并起来;
- (2) 它的一个限制是两个SQL语句所产生的栏位需要是同样的资料种类;
- (3) UNION只是将两个结果联结起来一起显示, 并不是联结两个表;
- (4) UNION在进行表链接后会筛选掉重复的记录。

UNION ALL:

- (1) 这个指令的目的也是要将两个 SQL 语句的结果合并在一起;
- (2) UNION ALL 和 UNION 不同之处在于 UNION ALL 会将每一个符合条件的资料都列出来, 无论资料值有无重复;
- (3) UNION ALL只是简单的将两个结果合并后就返回。这样, 如果返回的两个结果集中有重复的数据, 那么返回的结果集就会包含重复的数据了。

从效率上说, sql union all的执行效率要比sql union效率要高很多, 这是因为使用sql union需要进行排重, 而sql union All 是不需要排重的, 这一点非常重要, 因为对于一些单纯地使用分表来提高效率的查询, 完全可以使用sql union All。

补充: (摒弃的代码)

当时使用decode函数, 如果KWHD_WH_XZ='校级', 则输出自定义值'校级总数', 否则输出原始值; 同时通过group by 获取该列所有值, sum(decode(t.KWHD_WH_XZ,'校级',1,0))计算校级的个数。

```
select whxs1.num1 as 项目名称, whxs2.num2 as 数量
from
(select decode(KWHD_WH_XZ, '校级', '校级总数', KWHD_WH_XZ) as num1
from T_WSTB_KWHD_1 t
where KWHD_WH_XZ='校级'
group by KWHD_WH_XZ) whxs1,
(select sum(decode(t.KWHD_WH_XZ,'校级',1,0)) as num2
from T_WSTB_KWHD_1 t
where KWHD_WH_XZ='校级'
group by KWHD_WH_XZ ) whxs2;
```

输出如下, 但是再添加一行数据如何实现呢? 所以还是推荐UNION ALL。

信息	结果1
项目名称	数量
▶ 校级总数	168

4.表行列数据转换(表转置)

参考: http://blog.163.com/dreamman_yx/blog/static/26526894201121595846270

表名: table1

id	country	type	money
1	美国	A	70
2	美国	B	80
3	中国	A	60
4	中国	C	110
5	中国	C	150
6	美国	A	50
7	美国	C	100
8	中国	B	90

table2

country	A	B	C
中国	60	90	260
美国	120	80	100

转化规则就是根据 table1 国家来分组,计算出该国家同一 type 的 money 总和,呈现出来的样子就如表 table2
我在笔试的时候碰到的,当时没作出来,请教高人。

SQL语句如下:

```
select country, sum(case when type='A' then money end) as A,
sum(case when type='B' then money end) as B,
sum(case when type='C' then money end) as C
from table1
group by country
```

另一种方法源自文章: http://blog.sina.com.cn/s/blog_63772d910100pmln.html

方法介绍:

decode(条件,值1,结果1,值2,结果2,值3,结果3,... 值n,结果n,缺省值)

函数类比:

```
IF 条件=值1 THEN
    RETURN(结果1)
ELSIF 条件=值2 THEN
    RETURN(结果2)
    .....
ELSIF 条件=值n THEN
    RETURN(结果n)
ELSE
    RETURN(缺省值)
END IF
```

举个例子如下:

	NAME	RESULT
1	小李	负
2	小李	负
3	小李	胜
4	小明	胜
5	小明	胜
6	小明	胜
7	小明	负
8	小王	负

	姓名	胜	负
1	小明	3	1
2	小李	1	2
3	小王	0	1

SQL语句如下, 其中sum(decode(t.result,'胜',1,0))表示result字段如果值为“胜”, 则decode的结果值为1, 否则取缺省值0, 最后sum统计加和。

```
select
name as 姓名,sum(decode(t.result,'胜',1,0)) as 胜,sum(decode(t.result,'负',1,0)) as 负
from t_result t
group by name
order by 胜 desc,负 asc
```

最后希望文章对你有所帮助，其实SQL语句中还是有很多非常高深的变化，目前只窥得一二啊！fighting...O(∩_∩)O
(By:Eastmount 2016-01-22 深夜5点 <http://blog.csdn.net/eastmount/>)



Eastmount   博客专家

原创文章 462 获赞 6725 访问量 525万+

关注

他的留言板