

MySQL GROUP BY 语句

GROUP BY 语句根据一个或多个列对结果集进行分组。

在分组的列上我们可以使用 COUNT, SUM, AVG,等函数。

GROUP BY 语法

```
SELECT column_name, function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name;
```

实例演示

本章节实例使用到了以下表结构及数据，使用前我们可以先将以下数据导入数据库中。

```
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;

-- -----
-- Table structure for `employee_tbl`
-- -----
DROP TABLE IF EXISTS `employee_tbl`;
CREATE TABLE `employee_tbl` (
  `id` int(11) NOT NULL,
  `name` char(10) NOT NULL DEFAULT '',
  `date` datetime NOT NULL,
  `signin` tinyint(4) NOT NULL DEFAULT '0' COMMENT '登录次数',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `employee_tbl`
-- -----
BEGIN;
INSERT INTO `employee_tbl` VALUES ('1', '小明', '2016-04-22 15:25:33', '1'), ('2', '小王', '2016-04-20 15:25:47', '3'), ('3', '小丽', '2016-04-19 15:26:02', '2'), ('4', '小王', '2016-04-07 15:26:14', '4'), ('5', '小明', '2016-04-11 15:26:40', '4'), ('6', '小明', '2016-04-04 15:26:54', '2');
COMMIT;

SET FOREIGN_KEY_CHECKS = 1;
```

导入成功后，执行以下 SQL 语句：

```
mysql> set names utf8;
mysql> SELECT * FROM employee_tbl;
+----+-----+-----+-----+
| id | name  | date                | singin |
+----+-----+-----+-----+
| 1  | 小明 | 2016-04-22 15:25:33 | 1      |
| 2  | 小王 | 2016-04-20 15:25:47 | 3      |
| 3  | 小丽 | 2016-04-19 15:26:02 | 2      |
| 4  | 小王 | 2016-04-07 15:26:14 | 4      |
| 5  | 小明 | 2016-04-11 15:26:40 | 4      |
| 6  | 小明 | 2016-04-04 15:26:54 | 2      |
+----+-----+-----+-----+
6 rows in set (0.00 sec)
```

接下来我们使用 GROUP BY 语句 将数据表按名字进行分组，并统计每个人有多少条记录：

```
mysql> SELECT name, COUNT(*) FROM employee_tbl GROUP BY name;
+-----+-----+
| name  | COUNT(*) |
+-----+-----+
| 小丽 | 1        |
| 小明 | 3        |
| 小王 | 2        |
+-----+-----+
3 rows in set (0.01 sec)
```

使用 WITH ROLLUP

WITH ROLLUP 可以实现在分组统计数据基础上再进行相同的统计 (SUM,AVG,COUNT...)。

例如我们将以上的数据表按名字进行分组，再统计每个人登录的次数：

```
mysql> SELECT name, SUM(singin) as singin_count FROM employee_tbl GROUP BY name WITH ROLLUP;
+-----+-----+
| name  | singin_count |
+-----+-----+
| 小丽 | 2            |
| 小明 | 7            |
| 小王 | 7            |
| NULL  | 16           |
+-----+-----+
4 rows in set (0.00 sec)
```

其中记录 NULL 表示所有人的登录次数。

我们可以使用 coalesce 来设置一个可以取代 NULL 的名称，coalesce 语法：

```
select coalesce(a,b,c);
```

参数说明：如果a==null,则选择b；如果b==null,则选择c；如果a!=null,则选择a；如果a b c 都为null，则返回为null（没意义）。

以下实例中如果名字为空我们使用总数代替：

```
mysql> SELECT coalesce(name, '总数'), SUM(singin) as singin_count FROM employee_tbl GROUP BY name WITH ROLLUP;
```

coalesce(name, '总数')	singin_count
小丽	2
小明	7
小王	7
总数	16

```
4 rows in set (0.01 sec)
```

[← MySQL 导入数据](#)[MySQL UNION 操作符 →](#)**1 篇笔记**[写笔记](#)

1、group by 可以实现一个最简单的去重查询，假设有想看下有哪些员工，除了用 distinct,还可以用：

```
SELECT name FROM employee_tbl GROUP BY name;
```

返回的结果集就是所有员工的名字。

2、分组后的条件使用 HAVING 来限定，WHERE 是对原始数据进行条件限制。几个关键字的使用顺序为 where、group by、having、order by，例如：

```
SELECT name ,sum(*) FROM employee_tbl WHERE id<>1 GROUP BY name HAVING sum(*)>5 ORDER BY sum(*) DESC;
```

CAI 6个月前 (09-05)