

之前我们所涉及到的都是某列中的每个元素，或者整个列作处理，这次我们来看看对**局部数据**的分组处理。

一. GROUP BY

1. 使用背景：比如我要获取每个供应商所提供的产品数量，那么在我的返回列中，势必要有供应商-数目的形式，之前我们统计数目都是按照**整体**(COUNT())或者按照**某一个**特定的元素来的(WHERE), 这些都不符合要求，所以我们引入了GROUP BY

2. 使用示例：SELECT vend_id, COUNT(*) AS num FROM products GROUP BY vend_id

3. 如何理解：GROUP BY在指定列后，其他所有的函数，普通计算列——都要跟着每个**不同的**指定元素**重新计算**一遍！！比如说上述的COUNT(*)，其原本的意义在于对所有的列进行计算；但在指定GROUP BY的列以后，就是对每个元素进行相应的计算了。

4. 其它重点：

①GROUP BY**允许**嵌套使用

②GROUP BY中，**不允许**使用SELECT中指定的**别名**。

③GROUP BY要放到WHERE的后面，HAVING的前面（这点后面会细讲）

5. 好的习惯：

GROUP BY在使用时会对数据进行排序处理，但是不应该依赖于其自动排序的功能。也就是说：在GROUP BY后面最好**添加上ORDER BY子句**。

二. HAVING

如同可以**过滤数据**一样，Mysql还支持**过滤分组**。

HAVING子句的用法和WHERE相仿，要注意的一点是列都是针对指定元素计算而来的。

比如说：HAVING COUNT(*) >= 2；指对每个分组中计数过滤。

那么如何区别WHERE和HAVING呢？

可以理解为WHERE在分组前过滤数据，而HAVING 在分组后过滤数据。

(事实上，目前为止可以用WHERE的地方，都可以用HAVING来代替)

三. 小小的总结

目前已经讲了较多的SELECT了，其实SELECT中的子句都是由顺序的，下面顺便介绍一下这些子句的顺序：

SELECT--FROM--WHERE--GROUP BY--HAVING--ORDER BY--LIMIT

