* [1、概述](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m0)
* [2、原始表](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m1)
* [3、简单Group By](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m2)
* [4、Group By 和 Order By](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m3)
* [5、Group By中Select指定的字段限制](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m4)
* [6、Group By All](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m5)
* [7、Group By与聚合函数](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m6)
* [8、Having与Where的区别](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m7)
* [9、Compute 和 Compute By](http://www.cnblogs.com/rainman/archive/2013/05/01/3053703.html#m8)

**1、概述**

“Group By”从字面意义上理解就是根据“By”指定的规则对数据进行分组，所谓的分组就是将一个“数据集”划分成若干个“小区域”，然后针对若干个“小区域”进行数据处理。

**2、原始表**



**3、简单Group By**

**示例1**

select 类别, sum(数量) as 数量之和

from A

group by 类别

返回结果如下表，实际上就是分类汇总。



**4、Group By 和 Order By**

**示例2**

select 类别, sum(数量) AS 数量之和

from A

group by 类别

order by sum(数量) desc

返回结果如下表



在Access中不可以使用“order by 数量之和 desc”，但在SQL Server中则可以。

**5、Group By中Select指定的字段限制**

**示例3**

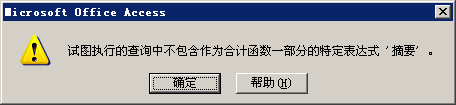
select 类别, sum(数量) as 数量之和, 摘要

from A

group by 类别

order by 类别 desc

示例3执行后会提示下错误，如下图。这就是需要注意的一点，在select指定的字段要么就要包含在Group By语句的后面，作为分组的依据；要么就要被包含在聚合函数中。



**6、Group By All**

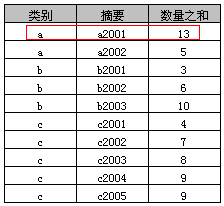
**示例4**

select 类别, 摘要, sum(数量) as 数量之和

from A

group by all 类别, 摘要

示例4中则可以指定“摘要”字段，其原因在于“多列分组”中包含了“摘要字段”，其执行结果如下表



“多列分组”实际上就是就是按照多列（类别+摘要）合并后的值进行分组，示例4中可以看到“a, a2001, 13”为“a, a2001, 11”和“a, a2001, 2”两条记录的合并。

SQL Server中虽然支持“group by all”，但[Microsoft SQL Server 的未来版本中将删除 GROUP BY ALL，避免在新的开发工作中使用 GROUP BY ALL](http://technet.microsoft.com/zh-cn/library/ms175028(v=sql.90).aspx)。Access中是不支持“Group By All”的，但Access中同样支持多列分组，上述SQL Server中的SQL在Access可以写成

select 类别, 摘要, sum(数量) AS 数量之和

from A

group by 类别, 摘要

**7、Group By与聚合函数**

在示例3中提到group by语句中select指定的字段必须是“分组依据字段”，其他字段若想出现在select中则必须包含在聚合函数中，常见的聚合函数如下表：

| **函数** | **作用** | **支持性** |
| --- | --- | --- |
| sum(列名) | 求和 |  |
| max(列名) | 最大值 |  |
| min(列名) | 最小值 |  |
| avg(列名) | 平均值 |  |
| first(列名) | 第一条记录 | 仅Access支持 |
| last(列名) | 最后一条记录 | 仅Access支持 |
| count(列名) | 统计记录数 | 注意和count(\*)的区别 |

**示例5：求各组平均值**

select 类别, avg(数量) AS 平均值 from A group by 类别;

**示例6：求各组记录数目**

select 类别, count(\*) AS 记录数 from A group by 类别;

**示例7：求各组记录数目**

**8、Having与Where的区别**

* where 子句的作用是在对查询结果进行分组前，将不符合where条件的行去掉，即在分组之前过滤数据，where条件中不能包含聚组函数，使用where条件过滤出特定的行。
* having 子句的作用是筛选满足条件的组，即在分组之后过滤数据，条件中经常包含聚组函数，使用having 条件过滤出特定的组，也可以使用多个分组标准进行分组。

**示例8**

select 类别, sum(数量) as 数量之和 from A

group by 类别

having sum(数量) > 18

**示例9：Having和Where的联合使用方法**

select 类别, SUM(数量)from A

where 数量 gt;8

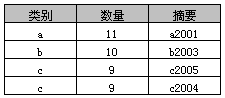
group by 类别

having SUM(数量) gt; 10

**9、Compute 和 Compute By**

select \* from A where 数量 > 8

执行结果：



**示例10：Compute**

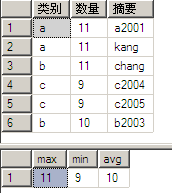
select \*

from A

where 数量>8

compute max(数量),min(数量),avg(数量)

执行结果如下：



compute子句能够观察“查询结果”的数据细节或统计各列数据（如例10中max、min和avg），返回结果由select列表和compute统计结果组成。

**示例11：Compute By**

select \*

from A

where 数量>8

order by 类别

compute max(数量),min(数量),avg(数量) by 类别

执行结果如下：



示例11与示例10相比多了“order by 类别”和“... by 类别”，示例10的执行结果实际是按照分组（a、b、c）进行了显示，每组都是由改组数据列表和改组数统计结果组成，另外：

* compute子句必须与order by子句用一起使用
* compute...by与group by相比，group by 只能得到各组数据的统计结果，而不能看到各组数据

在实际开发中compute与compute by的作用并不是很大，SQL Server支持compute和compute by，而Access并不支