

SQL进行排序、分组、统计的10个新技巧 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/259/2021_2022_SQL_E8_BF_9B_E8_A1_8C_E6_c98_259376.htm 以一种有意义的方式组织数据可能是一项挑战。有时你需要的可能是一个简单的排序，

但是通常你需要做更多，你需要分组来进行分析和统计。幸运的是，SQL提供了大量语句和操作来进行排序，分组和摘要。下面的一些技巧将会帮助你识别什么时候排序，什么时候分组，什么时候以及如何统计。对要每条语句和操作的详细信息请查看Books Online。

1.使用排序使数据有序 通常，你的所有数据真正需要的仅仅是按某种顺序排列。SQL的ORDER BY语句可以以字母或数字顺序组织数据。因此，相似的值按组排序在一起。然而，这个分组时排序的结果，并不是真的分组。ORDER BY显示每条记录而分组可能代表很多记录。

2.进行分组除去重复值 排序和分组之间的最大区别是：排序的数据显示所有记录（在限定标准范围之内），而分组数据不是显示所有记录。GROUP BY语句对于同样的值只显示一条记录。例如，下面的语句中的GROUP BY语句对数据源中重复出现的数据只返回唯一的zip编码列。

SELECT ZIP FROM Customers GROUP BY ZIP 只包括由GROUP BY和SELECT语句共同定义的那些记录，换句话说，SELECT列表必须满足GROUP BY列表，但是有一个例外就是SELECT列表可以包含聚合函数（GROUP BY语句不允许使用聚合函数）。需要注意的是GROUP BY语句不会对结果分组进行排序。为了使分组按字母或数字有序排列，需要添加ORDER BY语句。此外，在GROUP BY语句中不能引用使用

了别名的字段。分组栏目必须是潜在的数据，但它们并不需要显示在结果中。

3.在分组之前进行数据筛选 你可以添加一个WHERE语句来筛选有GROUP BY所得分组中的数据。例如，下面的语句只返回肯塔基州顾客的唯一ZIP编码列。

```
SELECT ZIP FROM Customers WHERE State = ' KY ' GROUP BY ZIP
```

必须注意的是WHERE语句是在GROUP BY语句求值之前进行数据过滤的。与GROUP BY语句一样，WHERE语句也不支持聚合函数。

4.返回所有分组 当你使用WHERE语句过滤数据时，结果分组中只显示你指定的那些记录，而符合分组定义但是不满足过滤条件的数据不会包含在某个分组中。当你想在分组中包含所有数据时添加关键字ALL即可，这时WHERE条件就不起作用。例如，在前面的例子中添加关键字ALL就会返回所有的ZIP分组，而不是仅在肯塔基州的那些。

```
SELECT ZIP FROM Customers WHERE State = ' KY ' GROUP BY ALL ZIP
```

这样看来，这两个语句存在冲突，你可能不会以这种方式使用关键字ALL。当你使用聚合函数计算某一列时，使用ALL关键字可能会很方便。例如，下面的语句计算每个肯塔基州ZIP中的顾客数，同时，还会显示其它的ZIP值。

```
SELECT ZIP, Count(ZIP) AS KYCustomersByZIP FROM Customers WHERE State = ' KY ' GROUP BY ALL ZIP
```

结果分组包括潜在数据中的所有ZIP值，然而，对于那些不是肯塔基州ZIP分组的聚合列(KYCustomersByZIP)将会显示0。

远程查询不支持 GROUP BY ALL。

5.分组后筛选数据 WHERE语句在GROUP BY语句之前进行计算。当你需要在分组之后筛选数据时，可以使用HAVING语句。通常情况下，WHERE语句和HAVING语句的返回结果是一样的，但是值得注意的

是这两个语句不可互换。当你迷惑时，可以遵循下面的说明：
：使用WHERE语句过滤记录，使用HAVING语句过滤分组。一般情况，你会使用HAVING语句和某个聚合函数计算一个分组。例如，下面的语句返回一个唯一的ZIP编码列，但是可能不会包含潜在数据源中所有的ZIP。 `SELECT ZIP, Count(ZIP) AS CustomersByZIP FROM Customers GROUP BY ZIP HAVING Count(ZIP) = 1` 只有那些包含一位顾客的分组显示在结果中。

6.进一步了解WHERE和HAVING语句 如果你对何时应该使用WHERE，何时使用HAVING仍旧很迷惑，请遵照下面的说明：WHERE语句在GROUP BY语句之前；SQL会在分组之前计算WHERE语句。HAVING语句在GROUP BY语句之后；SQL会在分组之后计算HAVING语句。

7.使用聚合函数统计分组数据 分组数据可以帮助我们分析数据，但是有时我们可能需要更多的信息而不仅仅是分组。你可以使用聚合函数来统计分组数据。例如，下面的语句显示每批订购单的总价钱。 `SELECT OrderID, Sum(Cost * Quantity) AS OrderTotal FROM Orders GROUP BY OrderID` 对于其它的分组来说，SELECT和GROUP BY列必须匹配。而SELECT语句包含聚合函数时这一规则是一个例外。

8.统计聚合数据 你可以继续统计数据为每个分组显示一个分类统计。SQL的ROLLUP操作符可以为每个分组显示一个额外的分类统计。这个分类统计是使用聚合函数计算每个分组中的所有记录得到的结果。下面的语句为每个分组计算OrderTotal：`SELECT Customer, OrderNumber, Sum(Cost * Quantity) AS OrderTotal FROM Orders GROUP BY Customer, OrderNumber WITH ROLLUP` 对于有两个分别为20和25 OderTotal值的分组，ROLLUP显示一

个OrderTotal值45。ROLLUP结果中的第一条记录是唯一的，因为它是计算所有分组记录，这个值是整个记录集的总值。ROLLUP在聚合函数中不支持DISTINCT，也不支持GROUP BY ALL语句。9.统计每个列 CUBE操作符比ROLLUP更进一步，它返回每个分组中重复值的个数。它的结果和ROLLUP相同，但是对每位客户的每一列CUBE包含一个额外的记录。下面的语句显示每个分组的统计和额外每位客户的统计。

```
SELECT Customer, OrderNumber, Sum(Cost * Quantity) AS  
OrderTotal FROM Orders GROUP BY Customer, OrderNumber  
WITH CUBE
```

CUBE可以给最综合的统计。它不仅完成聚合和ROLLUP的功能，还可以计算定义分组的其它列，换句话说，CUBE统计每个可能的列组合。CUBE不支持GROUP BY ALL语句。

10：对统计结果排序 当CUBE的结果令人迷惑时（它经常是这样），可以添加一个GROUPING函数，如下所示：

```
SELECT GROUPING(Customer), OrderNumber, Sum(Cost *  
Quantity) AS OrderTotal FROM Orders GROUP BY Customer,  
OrderNumber WITH CUBE
```

结果中每行包含两个额外的值：值1表示左边的值是一个统计值，是ROLLUP或CUBE的操作符。值0表示左边的值是一条由最初的GROUP BY语句产生的详细记录。

100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com