



查询的处理子句



1

查询中的排序

2

聚合函数的应用

3

查询中实施分组

4

查询中实施分类汇总

5

保存查询结果为表



数据排序的概念

- ☑ SELECT语句返回的查询数据集中的记录通常是按表中记录的物理顺序排列的。
- ☑ 可通过排序手段来改变查询数据集中记录的显示顺序，也就是对数据排序的意义。



实现数据排序的子句

- ☑ 通过在SELECT语句中使用ORDER BY子句对查询数据集的记录按照一个或多个列的值进行排序输出。排序依照的列又称为排序关键字。
- ☑ ORDER BY子句的格式为：
- ☑ ORDER BY { <排序表达式> [ASC | DESC] } [, ... n]



实现数据排序的子句

- ☑ 如果ORDER BY子句后是一个列名表达式表，则系统将根据各列的次序决定排序的优先级，然后排序。
- ☑ 如果指定了SELECT DISTINCT（去重复行），那么ORDER BY子句中的列名就必须出现在SELECT子句的列表中。



实现数据排序的子句

案例1：查询学生信息表“计算机软件1班”的同学信息，按性别升序，出生日期降序排列。

T-SQL代码：

Use student

GO

select *

from 学生信息

where 班级='计算机软件班'

order by 性别 asc, 出生日期 desc

	学号	姓名	性别	出生日期	系号	班级
1	0023	李雷	男	2002-03-05	001	计算机软件1班
2	0035	王运超	男	2001-06-09	001	计算机软件1班
3	0021	王广森	男	2000-01-02	001	计算机软件1班
4	0037	韩梅梅	女	2000-09-10	001	计算机软件1班
5	0036	张静静	女	1999-09-01	001	计算机软件1班
6	0022	杨阳	女	1999-02-06	001	计算机软件1班



查询的处理子句

1

查询中的排序

2

聚合函数的应用

3

查询中实施分组

4

查询中实施分类汇总

5

保存查询结果为表



聚合函数的概念

- ☑ 聚合函数也称为聚集函数或分组函数，这类函数将数据表在某一字段列上的值或与列相关的表达式值作为一个整体，进行某些特定的运算，或对一组记录中指定的列值或与列相关的表达式值进行某些特定的运算，并返回单个的数值做为函数值。
- ☑ 函数参数可以为字段列名、常量或由算术运算符连接而成的表达式。函数返回值作为新的一列出现在查询结果集中。



聚合函数的用法

- ☑ 聚合函数最简单的用法是放在SELECT语句的<目标项列表>中。
- ☑ 聚合函数的返回值一般为数值类型，常作为一个数据列项出现在查询结果列表中。
- ☑ 在这类用法中，为改善查询结果显示的效果，通常使用别名来命名聚合函数



主要的聚合函数列表

函数名	功能	应用格式
COUNT	计算并返回记录个数	COUNT(*)
	计算并返回指定列或表达式中值项的个数	COUNT([ALL DISTINCT] <列名/表达式>)
SUM	计算并返回指定列或表达式中值项的总和	SUM([ALL DISTINCT]<列名/表达式>)
AVG	计算并返回指定列或表达式中值项的平均值	AVG([ALL DISTINCT]<列名/表达式>)
MAX	计算并返回指定列或表达式中值项的最大值	MAX([ALL DISTINCT]<列名/表达式>)
MIN	计算并返回指定列或表达式中值项的最小值	MIN([ALL DISTINCT]<列名/表达式>)
VAR	计算并返回给定列或表达式值的统计方差	VAR([ALL DISTINCT]<列名/表达式>)
STDEV	计算并返回给定列或表达式值的统计标准偏差	STDEV([ALL DISTINCT]<列名/表达式>)

聚合函数的参数中，**ALL**关键字表示函数对指定列的所有值进行统计和计算，**DISTINCT**关键字说明函数仅对指定列的唯一值（不计重复值）进行统计和计算，**ALL**为默认设置。



聚合函数的用法

案例2：统计学生信息表中的男生人数。

T-SQL代码：

```
Use student
```

```
GO
```

```
select COUNT(*) 男生人数  
from 学生信息  
where 性别='男'
```

结果		消息
男生人数		
1	12	



聚合函数的用法

案例3：从学生信息表中统计出班级个数，每个班级只统计一次。

T-SQL代码：

Use student

GO

```
select count(distinct 班级) 班级数量  
from 学生信息
```

结果		消息
班级数量		
1	19	



聚合函数的用法

案例4：从学生成绩表中统计出学习了“001”号课程的学生人数，总成绩，平均成绩，最高分，最低分，标准差，方差。

T-SQL代码：

```
select COUNT(学号) 学生数, SUM(成绩) 总成绩, AVG(成绩) 平均成绩,  
MAX(成绩) 最高分, MIN(成绩) 最低分, STDEV(成绩) 标准差, VAR(成绩) 方差  
from 学生成绩  
where 课程号='001'
```

	学生数	总成绩	平均成绩	最高分	最低分	标准差	方差
1	7	506	72.2857142857143	90	58	16.2758951183879	264.904761904762



查询的处理子句

1

查询中的排序

2

聚合函数的应用



3

查询中实施分组

4

查询中实施分类汇总

5

保存查询结果为表



对查询数据集分组的概念

- ☑ GROUP BY子句则根据一系列或多列的值对查询数据集进行分组，将数据在指定列上具有相同值的记录划分在一起，组成一组。
- ☑ 对查询数据集分组的目的是为了细化聚合函数的作用对象，并借助聚合函数对每一组的记录进行统计，从而最终生成汇总的数据集。



GROUP BY 子句

- ☑ GROUP BY子句的语法格式如下：
- ☑ GROUP BY [ALL] <分组表达式> [,...n] [WITH { CUBE | ROLLUP }] [HAVING <条件表达式>]



无HAVING子句的分组查询

- ☑ GROUP BY子句用来对查询操作实施分组处理，HAVING子句则设置筛选条件。
- ☑ GROUP BY子句可以不带有HAVING条件，此时，查询到的整个数据集都将被分组，每一组在指定的一列或多列的组合上具有相同的值。



无HAVING子句的分组查询

案例5：从学生信息表中统计出各专业的学生数。要求查询结果显示专业名称和人数两列。

T-SQL代码：

```
select 专业, COUNT (学号) 人数  
from 学生信息  
group by 专业
```

	专业	人数
1	计算机软件技术	6
2	计算机网络技术	1
3	计算机应用技术	5
4	建筑设计	2



无HAVING子句的分组查询

案例6：统计学生信息表中的男女生人数。

T-SQL代码：

```
select 性别, COUNT(学号) 人数  
from 学生信息  
group by 性别
```

	性别	人数
1	男	12
2	女	16



带HAVING子句的分组查询

- ☑ HAVING子句用来为GROUP BY分组的结果设置筛选条件，使满足限定条件的那些组被挑选出来，构成最终的查询结果集。
- ☑ HAVING子句对组的筛选类似于WHERE子句对记录行的过滤。
- ☑ HAVING子句允许在<条件表达式>中包含聚合函数。聚合函数不能放在WHERE子句的逻辑表达式中



带HAVING子句的分组查询

案例7：从学生信息表中统计出各专业的学生数。要求查询结果显示专业名称和人数两列，且只显示专业人数大于5的。

T-SQL代码：

```
select 专业, COUNT (学号) 人数  
from 学生信息  
group by 专业  
having COUNT (学号) >= 5
```

	专业	人数
1	计算机软件技术	6
2	计算机应用技术	5



带HAVING子句的分组查询

案例8：从学生成绩表中找出学习了三门课以上的学生。要求查询结果显示学号和课程数。

T-SQL代码：

```
select 学号, COUNT(学号) 门数  
from 学生成绩  
group by 学号  
having COUNT(学号) >= 3
```

	学号	门数
1	0002	3
2	0022	3
3	0035	3



带HAVING子句的分组查询

思考题1:找出不及格门数超过三门课以上的学生。要求查询结果显示学号和补考课程数。



查询的处理子句

1

查询中的排序

2

聚合函数的应用

3

查询中实施分组

4

查询中实施分类汇总

5

保存查询结果为表





分类汇总的概念

- ☑ COMPUTE子句用于对查询数据集中的记录进行分类汇总，并产生一个额外的汇总记录行，插入到查询结果数据集的末尾。
- ☑ COMPUTE子句通过聚合函数，对列或表达式进行统计计算。



COMPUTE子句

- ☑ COMPUTE子句的语法格式如下:
- ☑ [COMPUTE { { SUM | AVG |
COUNT | MAX | MIN | VAR |
STDEV } (<列名/表达式>) } [,...n] [
BY { <分类表达式> } [,...n]]



无BY短语的COMPUTE子句

- ☑ SELECT语句中不带BY短语的COMPUTE子句，仅对当前查询数据集中的所有记录进行统计并形成一条附加的汇总记录。
- ☑ 不带BY短语的COMPUTE子句并不要求必须对记录集进行排序或分组。



无BY短语的COMPUTE子句

案例9：对学生成绩表中的成绩进行统计，统计所有学生考试的总成绩和平均成绩。

T-SQL代码：

```
select *  
from 学生成绩  
compute count(学号), sum(成绩), avg(成绩)
```

结果		消息	
	学号	课程号	成绩
1	0002	001	59
2	0002	005	75
3	0002	011	78
4	0003	012	89
5	0003	013	95
6	0004	001	60
7	0004	012	59
	cnt	sum	avg
1	24	1612	67.1666666666667



带BY短语的COMPUTE子句

- ☑ 带有BY短语的COMPUTE子句通常与ORDER BY子句一起使用，按照排序字段对所有记录进行分类，然后使用聚合函数，对不同类别的记录集进行统计运算，形成各个类别的汇总记录，并附加到各个类别记录集的明细列表尾部。
- ☑ COMPUTE子句计算聚合值，并且可以依然保持原有的值，新的聚合值作为特殊的列出现。



无BY短语的COMPUTE子句

案例10：对学生成绩表中的每个学生统计他所学课程的门数，总成绩和平均成绩。

T-SQL代码：

```
select *  
from 学生成绩  
order by 学号  
compute count(学号), sum(成绩), avg(成绩)  
by 学号
```

	cnt	sum	avg
1	3	212	70.66666666666667

学号	课程号	成绩	
1	0003	012	89
2	0003	013	95

	cnt	sum	avg
1	2	184	92

学号	课程号	成绩	
1	0004	001	60
2	0004	012	59

✓ 查询已成功执行。



查询的处理子句

1

查询中的排序

2

聚合函数的应用

3

查询中实施分组

4

查询中实施分类汇总

5

保存查询结果为表





保存查询结果的子句

- ☑ 查询结果不仅可以显示查看，还可以作为一个数据表永久保存起来。
- ☑ INTO子句用来将查询结果数据集保存到指定名称的数据表中。
INTO子句的语法格式如下：



INTO子句的语法格式

- ☑ INTO子句的语法格式如下：
- ☑ INTO <新表名>

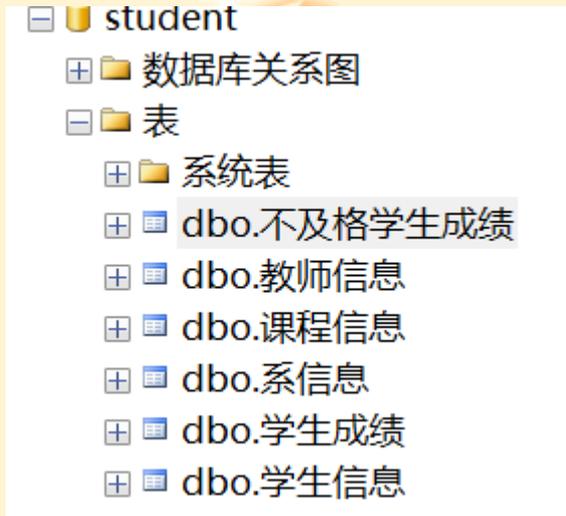


INTO子句的语法格式

案例11：查询学生中成绩小于60分的课程号并将查询结果以新表的形式建立。

T-SQL代码：

```
SELECT 学号, 课程号, 成绩  
INTO 不及格学生成绩  
FROM 学生成绩  
WHERE 成绩<60  
ORDER BY 学号
```





思考题

思考题2: 按照学号将数据集进行分组，然后分别统计各个组的平均成绩，并显示平均成绩大于60分的学生的学号和平均成绩。

思考题3: 查询学生中成绩小于75分的课程号、成绩以及这些学生单人成绩的总和。

思考题4: 查询平均成绩大于60分的这部分学生的平均成绩以及他们的总平均成绩。



下一章再见!