数据库系统实验七

18340013 陈琮昊

一、实验目的：

1.认识`NULL`值在数据库中的特殊含义。

2.了解`SQL`对空值和空集的处理。

3.熟练使用`SQL`语句进行对空置、空集相关的操作。

二、实验内容：

通过实验验证`SQL SERVER`对`NULL`的处理，包括:

(1)在查询的目标表达式中包含空值的运算。

(2)在查询条件中空值与比较运算符的运算结果。

(3)使用“`IS NULL`”或“`IS NOT NULL`”来判断元组该列是否为空值。

(4)对存在取空值的列按值进行`ORDER BY`排序。

(5)使用保留字`DISTINCT`对空值的处理。

(6)使用`GROUP BY`对存在取空值的属性值进行分组。

(7)结合分组考察空值对各个集合函数的影响，特别注意对`COUNT(*)`和`COUNT(列名)`的不同影响。

(8)考察结果集是空集时，各个集函数的处理情况。

(9)验证嵌套查询中返回空集的情况下与各个谓词的运算结果。

(10)进行与空值有关的等值连接运算。

三、实验题目：

(1)通过查询选修课程`C++`的学生的人数，其中成绩合格的学生人数，不合格的学生人数，讨论`NULL`值的特殊含义。

(2)查询选修课程`C++`的学生的编号和成绩，使用`ORDER BY`按成绩进行排序时，取`NULL`的项是否出现在结果中？如果有，在什么位置?

(3)在上面的查询的过程中，如果加上保留字`DISTINCT`会有什么效果呢?

(4)按年级对所有的学生进行分组，能得到多少个组?与现实的情况有什么不同?

(5)结合分组，使用集合函数求每个课程选修的学生的平均分，总的选课记录数，最高成绩，最低成绩，讨论考察取空值的项对集合函数的作用的影响。

(6)采用嵌套查询的方式，利用比较运算符和谓词ALL的结合来查询表`STUDENTS`中最晚入学的学生年级。当存在`GRADE`取空值的项时，考虑可能出现的情况，并解释。

四、实验过程与结果：

(1)

首先查询不及格的人数：

```
select COUNT(sid)
from CHOICES,COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++' and CHOICES.score<60
```

结果 消息

	(无列名)
1	724

然后查询及格的人数：

```
select COUNT(sid)
from CHOICES,COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++' and CHOICES.score>=60
```

结果 消息

	(无列名)
1	4817

然后查询 `score` 为 `NULL` 的人数：

```
select COUNT(sid)
from CHOICES,COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++' and CHOICES.score is null
```

结果 消息

	(无列名)
1	490

最后查询一下总人数：

```
select COUNT(sid)
from CHOICES,COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
```

结果 消息

	(无列名)
1	6031

可以发现前三者相加恰好等于总人数，可以看到`null`并不等价于数值0，它只是独立的一个运算，用`is null`进行查询。

(2)

```
select sid,score
from CHOICES,COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
order by score
```

结果 消息

	sid	score
1	845947855	NULL
2	890918686	NULL
3	898137922	NULL
4	867715893	NULL
5	872519782	NULL
6	886109186	NULL
7	884877167	NULL
8	893869264	NULL
9	834918408	NULL
10	873557912	NULL
11	802605966	NULL
12	878863812	NULL
13	812437303	NULL
14	824027469	NULL
15	849813323	NULL
16	878672282	NULL

可以看到取`NULL`的项出现在结果中，排在结果的最前面。

(3)

```
select distinct score,sid
from CHOICES,COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
order by score
```

结果 消息

	score	sid
1	NULL	800554079
2	NULL	800575453
3	NULL	800579690
4	NULL	800758208
5	NULL	800895421
6	NULL	800989091
7	NULL	801045247
8	NULL	801095737
9	NULL	801197660
10	NULL	801259696
11	NULL	801863111
12	NULL	802566342
13	NULL	802605966
14	NULL	802777255
15	NULL	802967609
16	NULL	803258105

查询已成功执行。 (local) (10.50 RTM) D42_56\Administrator (52) School 00:00:00 6024 行

可以看到此时`distinct`的应该是`sid`，因为都取`NULL`的`score`并未被合并。

```
select distinct score
from CHOICES,COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
order by score
```

结果 消息

	score
1	NULL
2	50
3	51
4	52
5	53
6	54
7	55
8	60
9	61
10	62
11	63
12	64
13	65
14	66
15	67
16	68

查询已成功执行。 (local) (10.50 RTM) D42_56\Administrator (52) School 00:00:00 47 行

如果只对`score`采取`distinct`，可以看到这时都取`NULL`的项被合并为一项了。

(4)

```
select distinct grade
from STUDENTS
order by grade
```

结果 消息

	grade
1	NULL
2	1991
3	1992
4	1993
5	1994
6	1995
7	1996
8	1997
9	1998
10	1999
11	2000
12	2001
13	2002
14	2003
15	2004
16	2020

查询已成功执行。 (local) (10.50 RTM) D42_56\Administrator (52) School 00:00:00 16 行

可以看到有16个组，但是这里面包含了`NULL`项。因此现实里应该是15项：

```
select distinct grade
from STUDENTS
where grade is not null
order by grade
```

结果 消息

	grade
1	1991
2	1992
3	1993
4	1994
5	1995
6	1996
7	1997
8	1998
9	1999
10	2000
11	2001
12	2002
13	2003
14	2004
15	2020

查询已成功执行。 (local) (10.50 RTM) D42_56\Administrator (52) School 00:00:00 15 行

(5)

```
select cid,AVG(score) as 'avgscore',COUNT(sid) as 'number',MAX(score) as 'maxscore',MIN(score) as 'minscore'
from CHOICES
group by cid
```

结果 消息

	cid	avgscore	number	maxscore	minscore
1	10008	75	5985	99	50
2	10019	75	6074	99	50
3	10018	76	5969	99	50
4	10040	75	6102	99	50
5	10011	76	6086	99	50
6	10028	75	6042	99	50
7	10035	76	6104	99	50
8	10021	76	5916	99	50
9	10046	76	6069	99	50
10	10032	75	6073	99	50
11	10005	76	6031	99	50
12	10050	75	6077	99	50
13	10034	76	6000	99	50
14	10015	75	5990	99	50
15	10036	75	6043	99	50
16	10045	76	6041	99	50

查询已成功执行。 (local) (10.50 RTM) D42_56\Administrator (52) School 00:00:00 50 行

`avgscore`代表平均分，`number`代表选课记录总数，`maxscore`、`minscore`分别代表最高分、最低分。

为了对比，去掉`sid`为`null`和`score`为`null`的项，即运行如下代码观察结果：

```sql
select cid,AVG(score) as 'avgscore',COUNT(sid) as 'number',MAX(score) as 'maxscore',MIN(score) as 'minscore'
from CHOICES
where score is not null and sid is not null
group by cid
```

结果 消息

	cid	avgscore	number	maxscore	minscore
1	10008	75	5512	99	50
2	10019	75	5555	99	50
3	10018	76	5520	99	50
4	10040	75	5596	99	50
5	10011	76	5621	99	50
6	10028	75	5567	99	50
7	10035	76	5607	99	50
8	10021	76	5462	99	50
9	10046	76	5606	99	50
10	10032	75	5614	99	50
11	10005	76	5541	99	50
12	10050	75	5609	99	50
13	10034	76	5506	99	50
14	10015	75	5508	99	50
15	10036	75	5547	99	50
16	10045	76	5549	99	50

对比两个结果可以发现，`avgscore`、`maxscore`、`minscore`并没有发生变化，但`number`一项发生了变化。可以看到去掉`NULL`后`number`项的结果要少了一些。也就是说`COUNT`是统计`NULL`的，`avg`、`max`、`min`函数是不统计`NULL`的。

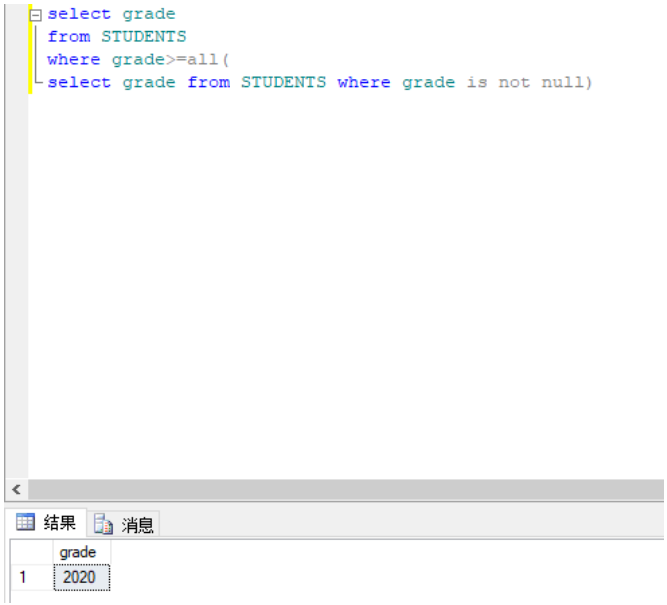
(6)

```sql
select grade
from STUDENTS
where grade>=all(
select grade from STUDENTS)
```

结果 消息

grade

发现结果为空，然后运行如下代码：



可以看到显示了我们想要的正确结果。说明空值影响了我们的查询。

五、实验体会：

本次实验专门来介绍`NULL`的处理方法。通过这次实验可以看到`NULL`并不等价于数值0；在分组时取`NULL`的值被分到了一组；集合函数中除了`COUNT`计算时要把取空值的项计算进去，其他的集合函数都忽略了取空值的项。