

# Lecture 8 数据库系统实验

Yubao Liu (刘玉葆)

School of Data and Computer Science

Sun Yat-sen University

- 本节课提纲

- 实验目的
- 实验内容
- 实验示例
- 练习

- # 实验目的

认识NULL值在数据库中的特殊含义。

了解SQL对空值和空集的处理。

熟练使用SQL语句进行对空置、空集相关的操作。

## • 实验内容

通过实验验证 SQL SERVER对**NULL**的处理，包括：

- 在查询的目标表达式中包含空值的运算。
- 在查询条件中空值与比较运算符的运算结果。
- 使用 “ IS NULL” 或 “ IS NOT NULL” 来判断元组该列是否为空值。
- 对存在取空值的列按值进行 ORDER BY排序。
- 使用保留字 DISTINCT对空值的处理。
- 使用 GROUP BY对存在取空值的属性值进行分组。
- 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。
- 考察结果集是空集时，各个集函数的处理情况。
- 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。
- 进行与空值有关的等值连接运算。

- 实验示例

以 school数据库为例，在该数据库中存在4张表格，分别为

- students (sid, sname , email , grade)
- teachers (tid, tname , email , salary )
- courses (cid, cname , hour )
- choices (no, sid , tid , cid , score )

在数据库中，存在这样的关系，学生可以选择课程。一个课程对应一个教师。

在CHOICES表中保存学生的选课记录。

# 实验示例

1. 查询选修编号为10028课程的学生人数，其中成绩合格的学生人数，不合格的人数。

SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT COUNT (*)
2 FROM CHOICES
3 WHERE CID='10028'
```

结果 消息

(无列名)
6042

RTM | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:01 | 1 行

总人数6024 →

SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT COUNT (*)
2 FROM CHOICES
3 WHERE SCORE>=60 AND CID='10028'
```

结果 消息

(无列名)
4812

RTM | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 1 行

← 及格人数4812人

SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT COUNT (*)
2 FROM CHOICES
3 WHERE SCORE<60 AND CID='10028'
```

结果 消息

(无列名)
755

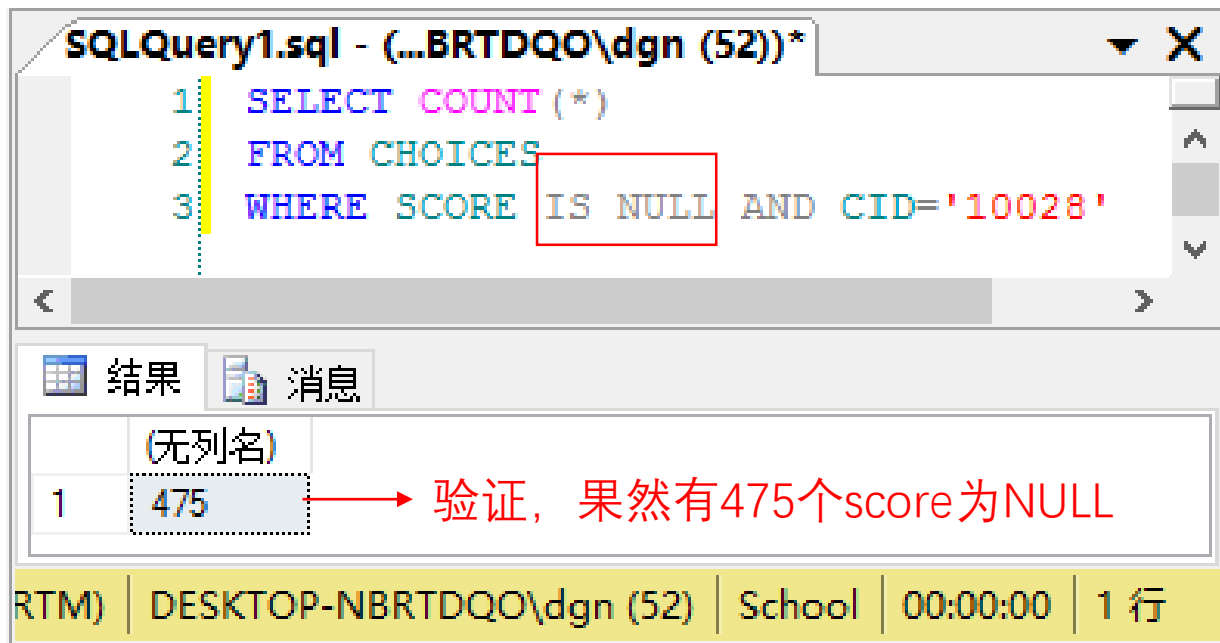
RTM | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 1 行

不及格755人 →

问题:  $6024 - 4812 - 755 = 475$   
还有475个人呢?

## • 实验示例

原因：当score取NULL值时，**NULL与所有的比较运算符都是不匹配的**，所以都不会出现在统计结果中。所以应该存在475个score为NULL的元组。以下为代码验证。



The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))\*". The query is as follows:

```
1 SELECT COUNT (*)
2 FROM CHOICES
3 WHERE SCORE IS NULL AND CID='10028'
```

The "IS NULL" part of the query is highlighted with a red box. Below the query, the "结果" (Results) tab is selected, showing a single row with the value 475. A red arrow points from the text "验证, 果然有475个score为NULL" to the value 475. The status bar at the bottom indicates "RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 1 行".

(无列名)
475

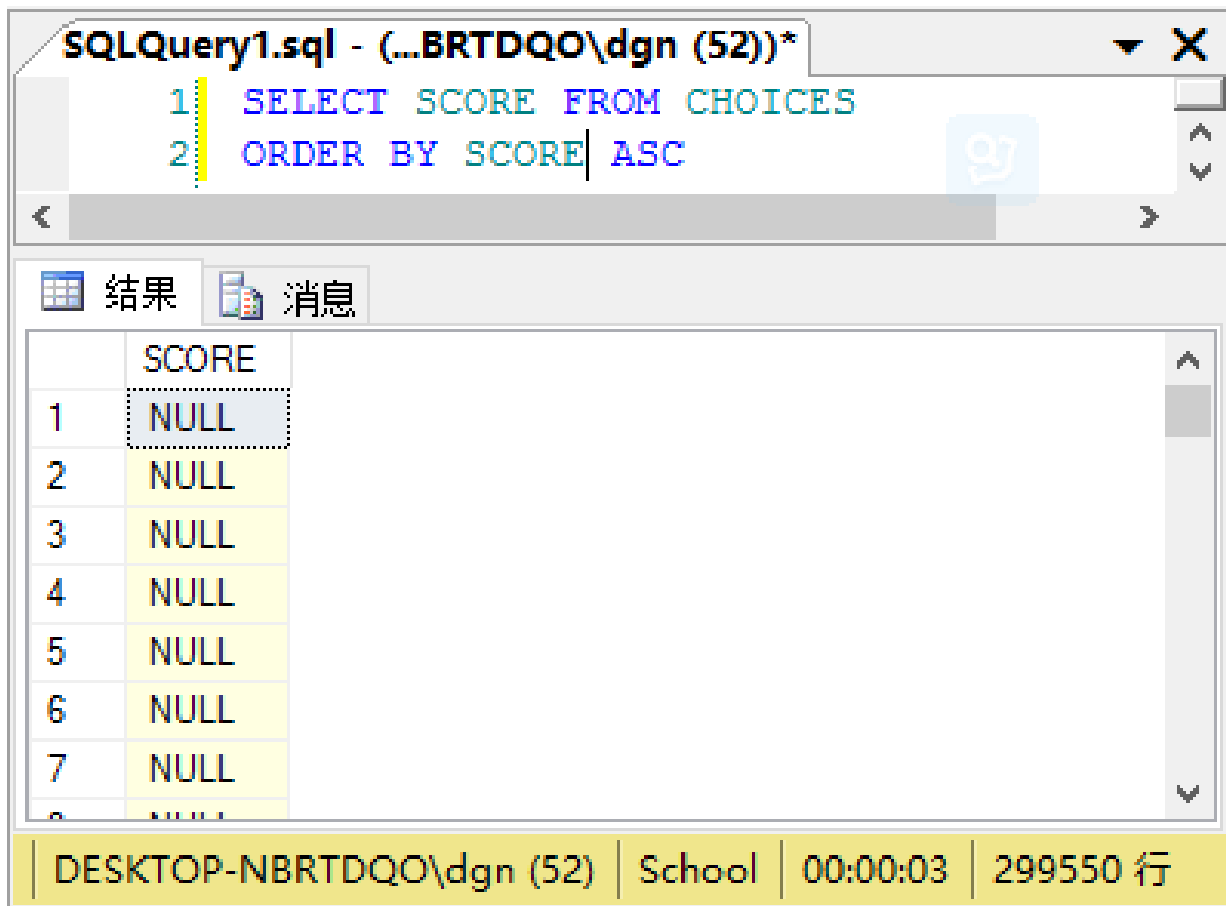
验证, 果然有475个score为NULL

RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 1 行

注意：判断是否为NULL,是“IS NULL”，而非“=NULL”

# 实验示例

2.将choices表按score排序，查看NULL值是否被忽略。



The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))\*". The query text is:

```
1 SELECT SCORE FROM CHOICES
2 ORDER BY SCORE ASC
```

Below the query, there are tabs for "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with the following data:

	SCORE
1	NULL
2	NULL
3	NULL
4	NULL
5	NULL
6	NULL
7	NULL
8	NULL

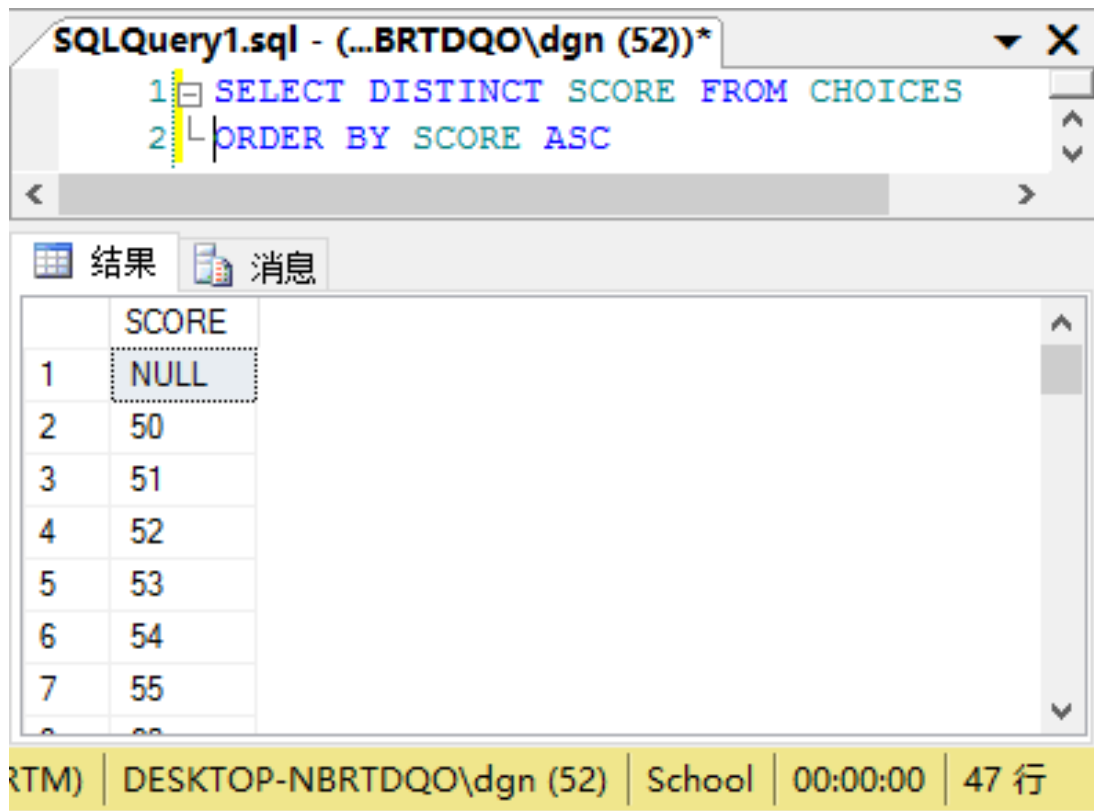
At the bottom of the window, a status bar displays: "DESKTOP-NBRTDQO\dgn (52) | School | 00:00:03 | 299550 行".

将成绩按从小到大排列，发现NULL值并没有被忽略，而是被当作最小值处理。



- 实验示例

3.如果上题查询加入了DISTINCT保留字，会有什么效果？



SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT DISTINCT SCORE FROM CHOICES
2 ORDER BY SCORE ASC
```

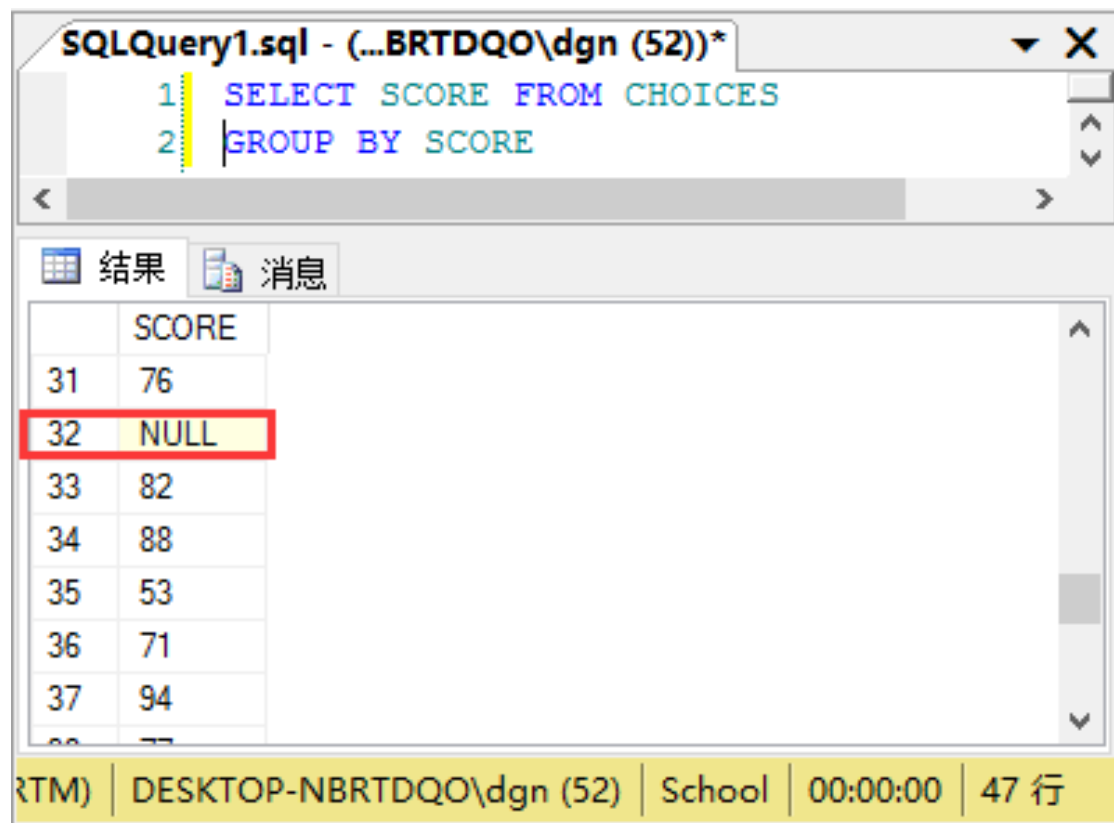
	SCORE
1	NULL
2	50
3	51
4	52
5	53
6	54
7	55

RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 47 行

所有取NULL值被看成是取一个相同值。

- 实验示例

4.通过实验考察GROUP BY对取值为NULL的项的处理。



The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))\*". The query text is:

```
1 SELECT SCORE FROM CHOICES
2 GROUP BY SCORE
```

Below the query, there is a tab labeled "结果" (Results) and a tab labeled "消息" (Messages). The "结果" tab is active, displaying a table with two columns: an index and "SCORE". The table contains the following data:

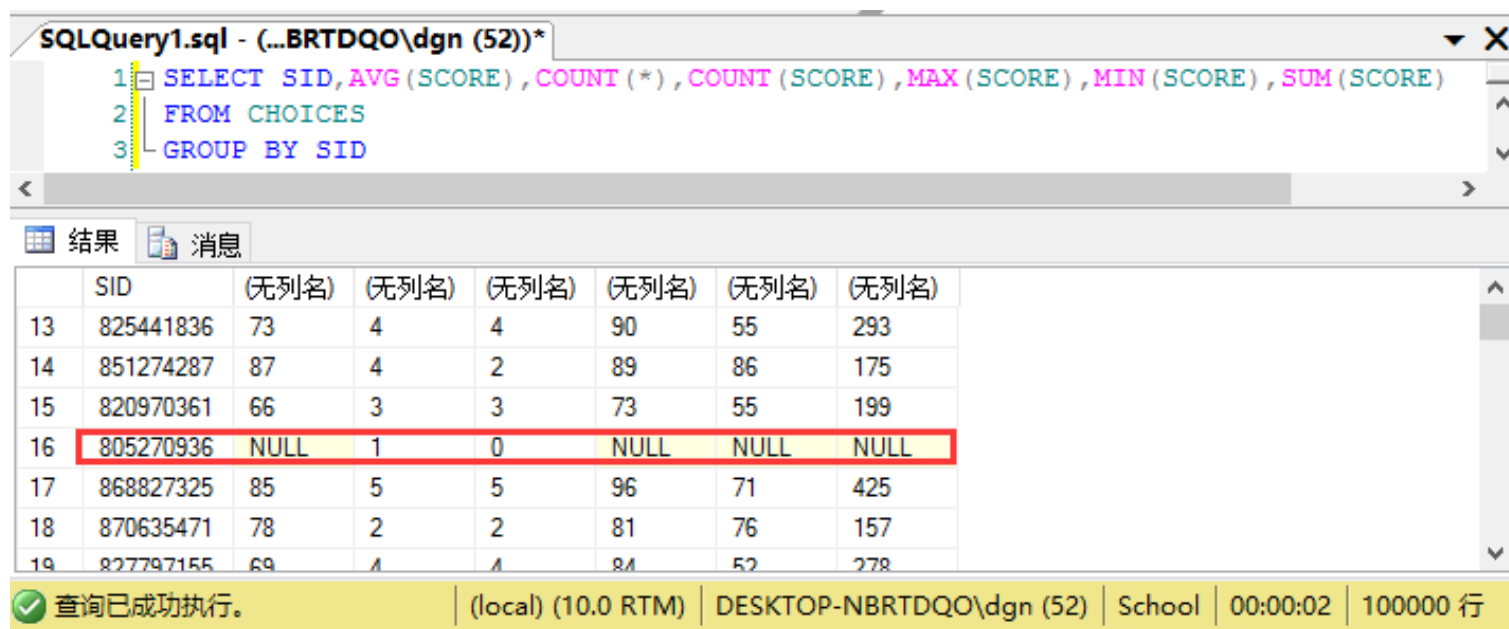
	SCORE
31	76
32	NULL
33	82
34	88
35	53
36	71
37	94

The row with index 32 and value NULL is highlighted with a red border. At the bottom of the window, the status bar shows "RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 47 行".

当查询结果存在取值为NULL的项时，分组时，取NULL值的会被当作一个分组。

# 实验示例

5.结合分组，使用集合函数求每个学生的平均分，总的选课记录数，最高成绩，最低成绩，总成绩。考察取空值的项对集合函数作用的影响。



SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT SID,AVG(SCORE),COUNT(*),COUNT(SCORE),MAX(SCORE),MIN(SCORE),SUM(SCORE)
2 FROM CHOICES
3 GROUP BY SID
```

结果 消息

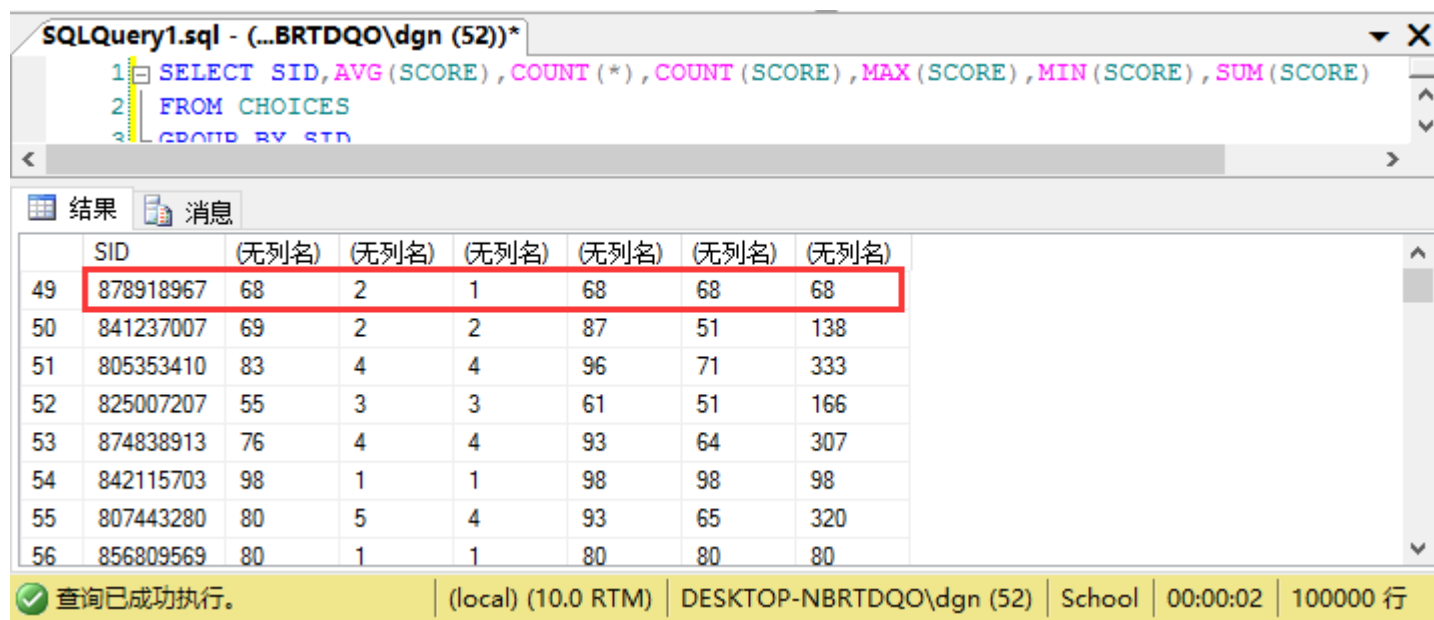
	SID	(无列名)	(无列名)	(无列名)	(无列名)	(无列名)	(无列名)
13	825441836	73	4	4	90	55	293
14	851274287	87	4	2	89	86	175
15	820970361	66	3	3	73	55	199
16	805270936	NULL	1	0	NULL	NULL	NULL
17	868827325	85	5	5	96	71	425
18	870635471	78	2	2	81	76	157
19	827797155	69	4	4	84	52	278

查询已成功执行。 | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:02 | 100000 行

这个学生有一个选课记录，然而成绩是NULL。在使用COUNT(\*)时，由于存在一个记录，所以取值为1。而对于COUNT(SCORE)，将取值为NULL的项忽略了。

## • 实验示例

我们还可以发现，编号为878918967 的学生有两个选课记录，一个成绩为68，一个为NULL,各个集合函数统计结果为(68,2,1,68,68,68),即再计算平均分，总分，最大，最小值的时候，取NULL的项被忽略了。



SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT SID,AVG(SCORE),COUNT(*),COUNT(SCORE),MAX(SCORE),MIN(SCORE),SUM(SCORE)
2 FROM CHOICES
3 GROUP BY SID
```

	SID	(无列名)	(无列名)	(无列名)	(无列名)	(无列名)	(无列名)
49	878918967	68	2	1	68	68	68
50	841237007	69	2	2	87	51	138
51	805353410	83	4	4	96	71	333
52	825007207	55	3	3	61	51	166
53	874838913	76	4	4	93	64	307
54	842115703	98	1	1	98	98	98
55	807443280	80	5	4	93	65	320
56	856809569	80	1	1	80	80	80

查询已成功执行。 (local) (10.0 RTM) DESKTOP-NBRTDQO\dgn (52) School 00:00:02 100000 行

总结：集合函数中，除了COUNT(\*)计算元组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。

# • 实验示例

6.查询成绩小于0的选课记录，统计总数，平均分，最大最小值。

SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT COUNT (*), AVG (SCORE), MAX (SCORE), MIN (SCORE)
2 FROM CHOICES
3 WHERE SCORE<0
```

<

结果 消息

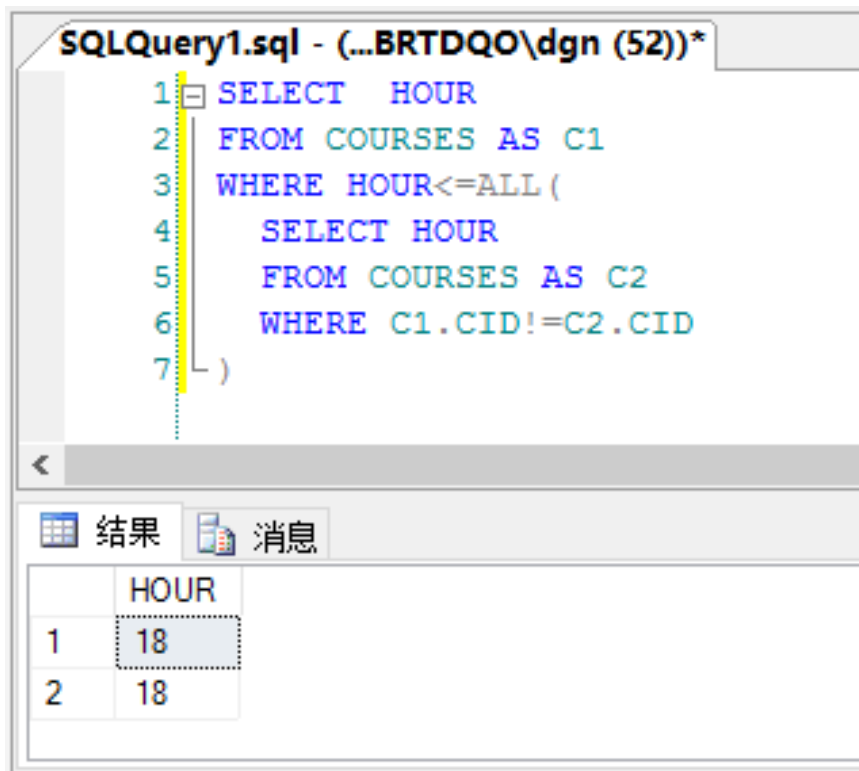
	(无列名)	(无列名)	(无列名)	(无列名)
1	0	NULL	NULL	NULL

因为数据库中所有成绩都不小于0，所以查询为空集。

对一个空集使用集合函数，除了COUNT(\*)返回0，其他均返回NULL.

## • 实验示例

7.使用以下代码查询COURSES表中最少的课时（利用比较运算符和谓词ALL）。  
假设数据库中只有一个记录，使用该方法会有什么结果？



```
SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 SELECT  HOUR
2 FROM    COURSES AS C1
3 WHERE   HOUR<=ALL (
4         SELECT HOUR
5         FROM    COURSES AS C2
6         WHERE   C1.CID!=C2.CID
7     )
```

	HOUR
1	18
2	18

该表有两条记录。

注意：假设数据库只有一个记录，即子查询中  
“C1.CID<>C2.CID” 无法成立,子查询返回为  
空集。SQL对于“ **HOUR<=ALL(空集)** ” 返回  
**TRUE**，所以上述语句也能成功查询到这唯一的  
元组。

# 实验示例

8.对于下面学生表和教师表，对这两个表作对**姓名的等值连接**运算,找出既是老师又是学生的人员的学生编号和教师编号.

学生表

	NO	SID	SNAME
1	1	0129871001	王小明
2	2	0129871002	李兰
3	3	0129871005	NULL
4	4	0129871004	关红

教师表

	NO	TID	TNAME
1	1	100189	王小明
2	2	1001801	李晓
3	3	100121	NULL
4	4	100128	NULL

SQLQuery3.sql - (...BRTDQO\dgn (54))\*

```
1 SELECT *
2 FROM T, S
3 WHERE T.TNAME=S.SNAME
```

结果 消息

	NO	TID	TNAME	NO	SID	SNAME
1	1	100189	王小明	1	0129871001	王小明

等值连接后，姓名为**NULL**的项都被忽略了。  
即，两个属于不同表，并且对应列都为  
NULL的列是不会发生连接的，也不会出现在结果中。

## • 练习

- (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。