

中山大学数据科学与计算机学院

移动信息工程专业-数据库系统

本科生实验报告

(2017-2018 学年秋季学期)

课程名称：数据库系统实验

教学班级	15M1	专业（方向）	移动互联网
学号	15352408	姓名	张镓伟

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

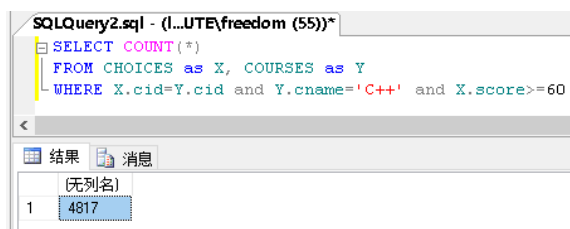
通过实验验证 SQL SERVER 对 NULL 的处理，包括：

- 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、实验过程及结果

- (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。

合格：



```
SQLQuery2.sql - (1...UTE\freedom (55))*
SELECT COUNT(*)
FROM CHOICES as X, COURSES as Y
WHERE X.cid=Y.cid and Y.cname='C++' and X.score>=60
```

	(无列名)
1	4817



不合格的情况:

```
SELECT COUNT(*)
FROM CHOICES as X, COURSES as Y
WHERE X.cid=Y.cid and Y.cname='C++' and X.score<60
```

结果	消息
(无列名)	
1	724

而选修总人数为 6031:

```
SQLQuery2.sql - (!...UTE\freedom (55))*
SELECT COUNT(*)
FROM CHOICES as X, COURSES as Y
WHERE X.cid=Y.cid and Y.cname='C++'
```

结果	消息
(无列名)	
1	6031

可以发现,合格人数+不合格人数=5541<6031,这是因为剩下的人的 score 是 NULL,与所有比较运算符都不匹配,它是作为一个独特的值而存在,意义在于没有记录:

```
SQLQuery2.sql - (!...UTE\freedom (55))*
SELECT COUNT(*)
FROM CHOICES as X, COURSES as Y
WHERE X.cid=Y.cid and Y.cname='C++' and score IS NULL
```

结果	消息
(无列名)	
1	490

加上 NULL 的人数 490 之后,刚好符合总人数。

- (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?

```
SELECT sid,score
FROM CHOICES as X, COURSES as Y
WHERE X.cid=Y.cid and Y.cname='C++'
ORDER BY score
```

sid	score
845947855	NULL
890918686	NULL
898137922	NULL
867715892	NULL

```
SELECT sid,score
FROM CHOICES as X, COURSES as Y
WHERE X.cid=Y.cid and Y.cname='C++'
ORDER BY score
```

sid	score
856062891	NULL
800895421	NULL
863153426	NULL
806782952	NULL
840545438	NULL
869887105	NULL
863905602	50
853508519	50

由图可知,取 NULL 的项会出现在结果中,而且是按照最小值处理排在最前面。

- (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。

```
SELECT DISTINCT sid,score
FROM CHOICES as X, COURSES as Y
WHERE X.cid=Y.cid and Y.cname='C++'
ORDER BY score
```

sid	score
899287584	NULL
899960738	NULL
899982217	NULL
801935884	50

查询已成功执行。 (local) (10.0 RTM) FREEDOM-COMPUTE\freedom... School 00:00:00 6024 行



可以看到总的查询记录数变少了,说明有的人同一门课重复考了几次。

- (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?

```
SELECT grade
FROM STUDENTS
GROUP BY grade
ORDER BY grade
```

grade
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

一共有 15 个分组,与现实不同的是,现实中不存在 grade=NULL 的年级

- (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。

算上空值:

```
SELECT cid,AVG(SCORE) as AverageScore,COUNT(*) as Total,MAX(score) as MaxScore,
MIN(score) as MinScore
FROM CHOICES
GROUP BY cid
```

cid	AverageScore	Total	MaxScore	MinScore
1	75	5985	99	50
2	75	6074	99	50
3	76	5969	99	50
4	75	6102	99	50
5	76	6086	99	50
6	75	6042	99	50

去掉空值:

```
SELECT cid,AVG(SCORE) as AverageScore,COUNT(*) as Total,MAX(score) as MaxScore,
MIN(score) as MinScore
FROM CHOICES
WHERE score IS NOT NULL
GROUP BY cid
```

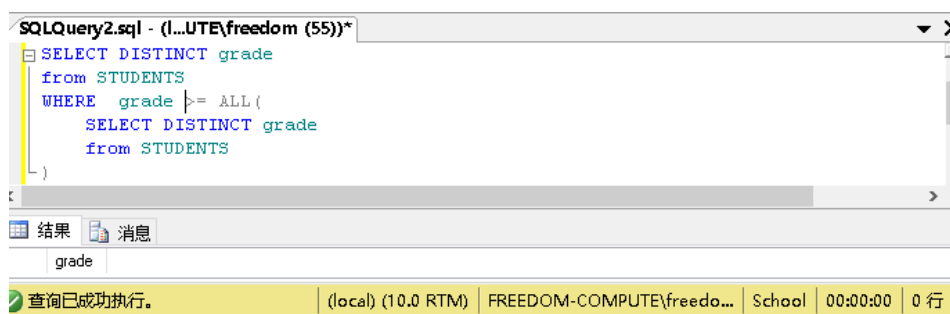
cid	AverageScore	Total	MaxScore	MinScore
1	75	5512	99	50
2	75	5555	99	50
3	76	5520	99	50
4	75	5596	99	50
5	76	5621	99	50
6	75	5567	99	50

对比两种情况发现, score 的空值会对选课总人数的统计造成影响,但是对平均分最高分和最低分的统计没有影响。

- (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。

取空值时:

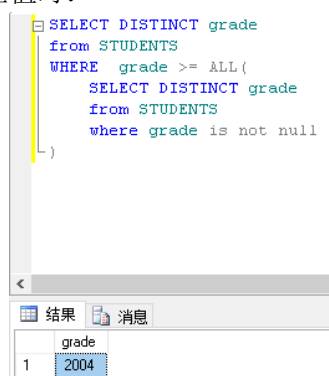


```
SQLQuery2.sql - (1...UTE\freedom (55))*  
SELECT DISTINCT grade  
from STUDENTS  
WHERE grade >= ALL(  
    SELECT DISTINCT grade  
    from STUDENTS  
)
```

Results: grade

查询已成功执行。 (local) (10.0 RTM) FREEDOM-COMPUTE\freedo... School 00:00:00 0 行

无空值时:



```
SELECT DISTINCT grade  
from STUDENTS  
WHERE grade >= ALL(  
    SELECT DISTINCT grade  
    from STUDENTS  
    where grade is not null  
)
```

Results: grade

grade
1 2004

可以看到,有空值时,查询没有结果,这是因为 NULL 不参与任何比较运算,所以我们没有办法找到一个 grade,它不小于嵌套查询的 grade 的集合中的所有元素,因为这个集合里有一个 NULL 值。

四、实验感想

这次实验的目的是让我们了解空值在数据库中的意义。我的理解就是它是一个特殊的值,意思是“没有记录”。这个值的存在还是很有意义的,因为我们不一定会知道所有完整的数据。但是要注意的是,当数据表中存在 NULL 时,我们一些涉及比较运算的 sql 语句就要将他们考虑进去了,否则有可能得到的结果是不对的。