

# 数据库实验报告 实验九 空值和空集的处理

姓名	学号	班级	课室
熊明	20305055	计科5班	D503

## 一、实验目的

认识NULL值在数据库中的特殊含义，了解空值和空集对于数据库的数据查询操作，特别是空值在条件表达式中与其他的算术运算符或逻辑运算符的运算中，空集作为嵌套查询的子查询的返回结果时候的特殊性，能够熟练使用SQL语句来进行与空值、空集相关的操作。

## 二、实验环境

数据库：Mysql

图形化工具：Navicat Premium 16

## 三、实验内容

通过实验验证数据库管理系统对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，区分数据库的多种取值与现实中的多种取值的不同。
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。

## 四、课内实验

1. 查询所有选课记录的成绩并将其换算为五分制(满分为5分，合格为3分),注意SCORE取NULL值的情况。

```
1 SELECT score*5/100 as score_5
2 FROM choices
3 WHERE score is NOT NULL
```

部分结果如下：

score_5
3.8000
4.3500
2.7000
4.6000
4.1000
3.8500
3.3500
4.5000
3.0000
3.5000
4.1000
3.8000
4.4000
3.6000
3.2000

2. 查询选修编号为10028的课程的学生的人数，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。

首先查看选修10028且有成绩的学生人数：

```
1 SELECT cid,COUNT(score) as students_num
2 FROM choices
3 WHERE score IS NOT NULL AND cid = 10028
4 GROUP BY cid
```

结果如下：

cid	students_num
10028	5567

然后进行筛选成绩合格和不合格的学生人数：

```
1 SELECT c1.cid,c1.passed,unpassed
2 FROM(
3     SELECT cid,COUNT(score) AS passed
4     FROM choices
5     WHERE score IS NOT NULL AND score >=60
6     GROUP BY cid) as c1
7 LEFT JOIN
8 (SELECT c2.cid,c2.unpassed
9 FROM(
10     SELECT cid,COUNT(score) AS unpassed
11     FROM choices
12     WHERE score IS NOT NULL AND score <60
13     GROUP BY cid) as c2
14 WHERE cid = 10028) AS cc
15 ON c1.cid = cc.cid
16 WHERE c1.cid = 10028
```

结果如下：

cid	passed	unpassed
▶ 10028	4812	755

在此问题中，null空值代表着该学生选修了cid为10028的课程，但是没有最终成绩。

3. 通过实验检验在使用ORDER BY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?

使用下面语句对choices表中score列进行排序操作：

```
1 SELECT score
2 FROM choices
3 ORDER BY score
```

结果如下：

score
(Null)
(Null)
(Null)
(Null)
(Null)
▶ (Null)
(Null)
50
50
50
50
50
50
50
50
50
50
50

可以看出，使用order by会将null值放在最前面

4. 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?

```
1 SELECT DISTINCT score
2 FROM choices
3 ORDER BY score
```

结果如下：

score
(Null)
50
51
52
53
54
55
60
61
62
63

所有成绩情况只会出现一次

##### 5. 通过实验说明使用分组GROUP BY对取值为NULL的项的处理

使用下面语句对choices表中score进行分组，并统计学生人数：

```
1 SELECT score,COUNT(sid)
2 FROM choices
3 GROUP BY score
```

得到的结果如下：

score	COUNT(sid)
76	6005
87	6060
54	6023
92	5931
82	6035
77	6085
67	6080
90	6020
60	6029
(Null)	23913
70	5824
88	5978

可以看出，group by会将空值都分到同一组

- count(\*)包括了所有的列，相当于行数，在统计结果的时候，不会忽略列值为NULL
- count(列名)只包括列名那一列，在统计结果的时候，会忽略列值为空（这里的空不是只空字符串或者0，而是表示null）的计数，即某个字段值为NULL时，不统计。

使用如下语句，统计相同cid下的分数个数：

```

1 SELECT *
2 FROM
3     (SELECT cid,COUNT(score)
4      FROM choices
5      GROUP BY cid) as c1
6 JOIN
7     (SELECT cid,COUNT(*)
8      FROM choices
9      GROUP BY cid) as c2
10 ON c1.cid = c2.cid

```

得到以下结果：

	cid	COUNT(score)	cid(1)	COUNT(*)
▶	10001	5435	10001	5898
	10002	5502	10002	6013
	10003	5493	10003	5975
	10004	5639	10004	6110
	10005	5541	10005	6031
	10006	5603	10006	6090
	10007	5494	10007	5965
	10008	5512	10008	5985
	10009	5487	10009	5965
	10010	5516	10010	6027
	10011	5621	10011	6086
	10012	5348	10012	5819
	10013	5474	10013	5982

可以看出，count(\*)的数量要比count(score)多，说明count(\*)在统计的时候不会忽略null  
对剩下集合函数的处理如下

1. sum函数：

```

1 SELECT cid,SUM(score)
2 FROM choices
3 GROUP BY cid

```

SUM() 函数在计算时，会忽略列值为 NULL 的行

cid	SUM(score)
▶ 10001	412910
10002	417571
10003	417155
10004	429290
10005	421211
10006	426220
10007	417060
10008	417033
10009	418175
10010	418264

2. avg函数:

```
1 SELECT cid,AVG(score)
2 FROM choices
3 GROUP BY cid
```

avg函数在使用的时候, 会忽略null的值

cid	AVG(score)
▶ 10001	75.9724
10002	75.8944
10003	75.9430
10004	76.1287
10005	76.0171
10006	76.0700
10007	75.9119
10008	75.6591
10009	76.2120
10010	75.8274

3. max, min函数:

都会忽略null值

```
1 SELECT cid,MIN(score), MAX(score)
2 FROM choices
3 GROUP BY cid
```

得到结果如下:

cid	MIN(score)	MAX(score)
10001	50	99
10002	50	99
10003	50	99
10004	50	99
10005	50	99
10006	50	99
10007	50	99
10008	50	99
10009	50	99
10010	50	99
10011	50	99
10012	50	99
10013	50	99

6. 结合分组，使用集合函数求每个同学的平均分，总的选课记录，最高成绩，最低成绩，总成绩。

代码如下：

```
1 SELECT AVG(score),COUNT(*),MAX(score),MIN(score),SUM(score)
2 FROM choices
3 GROUP BY sid
```

部分结果如下：

AVG(score)	COUNT(*)	MAX(score)	MIN(score)	SUM(score)
62.3333	3	67	60	187
73.6667	4	82	60	221
66.0000	1	66	66	66
89.0000	3	96	77	267
71.0000	5	84	55	284
74.0000	3	93	64	222
76.0000	1	76	76	76
88.5000	3	94	83	177
87.5000	3	95	80	175
75.5000	3	88	63	151
73.5000	2	79	68	147
84.5000	4	92	76	338
70.0000	3	88	53	210
71.0000	2	71	71	71
83.0000	2	83	83	83

7. 查询成绩小于0的选课记录，统计总数，平均分，最大值和最小值

## 五、自我实践

1. 查询所有课程记录的上课学时(数据库中为每星期学时),以一学期十八个星期计算每个课程的总学时, 注意HOUR取NULL值的情况。

```
1 SELECT c.cid,a*`hour`*18 AS among_hour
2 FROM
3 ( SELECT cid,COUNT(*) as a
4   FROM choices
5   GROUP BY cid) AS c
6 JOIN
7 (SELECT cid,hour
8  FROM courses) as cc
9 ON c.cid = cc.cid
```

部分结果如下:

cid	among_hour
▶ 10001	10191744
10002	9524592
10003	5162400
10004	5279040
10005	6513480
10006	5261760
10007	3221100
10008	6463800
10009	3865320
10010	5424300
10011	3943728
10012	4189680
10013	4953096

2. 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

```
1 SELECT count(score) as unpassed
2 FROM choices
3 WHERE cid = (SELECT cid FROM courses WHERE cname = 'C++') AND score <
4 60;
5 SELECT count(score) as passed
6 FROM choices
7 WHERE cid = (SELECT cid FROM courses WHERE cname = 'C++') AND score >=
8 60;
9 SELECT count(score) as among
10 FROM choices
11 WHERE cid = (SELECT cid FROM courses WHERE cname = 'C++');
```

结果如下:



unpassed
▶ 724

passed
▶ 4817

among
▶ 5541

3. 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?

```

1 SELECT sid,score
2 FROM choices
3 WHERE cid = (SELECT cid FROM courses WHERE cname = 'C++')
4 ORDER BY score

```

部分结果如下:

sid	score
864171732	(Null)
848049773	(Null)
819620160	(Null)
892661911	(Null)
826634334	50
892021253	50
▶ 811656358	50
803865142	50

可以看出，null值在最上面

4. 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?

```

1 SELECT DISTINCT score
2 FROM choices
3 WHERE cid = (SELECT cid FROM courses WHERE cname = 'C++')
4 ORDER BY score

```

部分结果如下:

score
(Null)
50
51
52
53
54
55
60

所有分数情况只会出现一次

5. 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?

```
1 SELECT grade,COUNT(sid)
2 FROM students
3 GROUP BY grade
```

按照年级分组，结果如下：

grade	COUNT(sid)
1992	6595
2002	6659
1995	6593
1997	6618
1999	6664
1991	6700
2000	6671
1994	6725
1996	6754
1998	6622
1993	6615
2001	6754
2004	6716

```
1 SELECT count(grade)
2 FROM(
3 SELECT grade,COUNT(sid)
4 FROM students
5 GROUP BY grade) as c
```

结果一共可以得到15个分组，多了一个null分组

	2004	6716
	(Null)	6686
	2003	6628
▶	1	1

6. 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。

```
1 SELECT AVG(score),SUM(score),MAX(score),MIN(score)
2 FROM choices
3 GROUP BY cid
```

部分结果如下：

	AVG(score)	SUM(score)	MAX(score)	MIN(score)
▶	75.9724	412910	99	50
	75.8944	417571	99	50
	75.9430	417155	99	50
	76.1287	429290	99	50
	76.0171	421211	99	50
	76.0700	426220	99	50
	75.9119	417060	99	50
	75.6591	417033	99	50
	76.2120	418175	99	50
	75.8274	418264	99	50
	76.3341	429074	99	50
	76.1281	407133	99	50

空值的项对集合函数的作用的影响见[5.通过实验说明使用分组GROUP BY对取值为NULL的项的处理](#)