数据库原理及应用实验报告

(2021 级)

实验题目 实验 6　SQL 的空值和空集处理

学生姓名 温家伟

学生学号 202103151422

学科(专业) 大数据分析 2101 班

所在学院 理学院

提交日期 2023 年 4 月 30 日

实验6、SQL的空值和空集处理

6.1 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者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。

6.2 实验内容

通过实验验证在数据库原理解析中分析过的DBMS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，区分数据库的多种取值与现实中的多种取值的不同。
- 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。

6.3 实验步骤

(1)查询所有选课记录的成绩并将它换算为五分制（满分为5分，合格为3分，即公式：$round(成绩/100*5)$），注意,创建表时允许`Score`取`NULL`值。

```
mysql> select Sno, Tno, Cno, round(score/100*5) from STC;
+-----+-----+-----+--------------------+
| Sno | Tno | Cno | round(score/100*5) |
+-----+-----+-----+--------------------+
| S01 | T01 | C01 |                  4 |
| S01 | T03 | C03 |                  4 |
| S02 | T01 | C01 |                  4 |
| S02 | T01 | C08 |                  4 |
| S02 | T02 | C02 |                  2 |
| S02 | T02 | C09 |                  4 |
| S02 | T03 | C03 |               NULL |
| S02 | T04 | C04 |               NULL |
| S02 | T04 | C06 |                  4 |
| S02 | T05 | C05 |                  4 |
| S02 | T05 | C07 |                  5 |
| S02 | T06 | C11 |                  4 |
| S02 | T07 | C10 |                  4 |
| S03 | T01 | C01 |                  4 |
| S03 | T01 | C08 |                  3 |
| S03 | T02 | C02 |                  5 |
| S04 | T05 | C05 |                  5 |
| S04 | T06 | C06 |                  4 |
| S26 | T04 | C04 |                  4 |
| S26 | T07 | C10 |                  2 |
| S52 | T01 | C08 |                  3 |
| S52 | T02 | C09 |                  4 |
| S52 | T05 | C05 |               NULL |
| S52 | T06 | C11 |                  5 |
| S52 | T07 | C10 |                  5 |
+-----+-----+-----+--------------------+
25 rows in set (0.00 sec)
```

(2)通过查询选修编号`C07`的课程的学生的人数，其中成绩合格的学生人数，不合格的人数，讨论`NULL`值的特殊含义。

> 为了讨论空值的特殊含义，特意插入一条空值。

```
mysql> select count(Cno) as 选修编号C07的课程的学生的人数 from STC where Cno = 'C07';
+-------------------------------------------+
| 选修编号C07的课程的学生的人数             |
+-------------------------------------------+
|                                         3 |
+-------------------------------------------+
1 row in set (0.00 sec)
```

```
mysql> select count(Cno) as 成绩合格的学生人数 from STC where Cno = 'C07' and Score >= 60;
+----------------------------+
| 成绩合格的学生人数         |
+----------------------------+
|                          1 |
+----------------------------+
1 row in set (0.00 sec)

mysql> select count(Cno) as 成绩不合格的学生人数 from STC where Cno = 'C07' and Score  < 60;
+------------------------------+
| 成绩不合格的学生人数         |
+------------------------------+
|                            1 |
+------------------------------+
1 row in set (0.00 sec)
```

> 因为空值不参与运算，所以`Score<60`和`Score>=60`都不会把空值选出来。

(3)通过实验检验在使用`ORDER BY`进行排序时(升序，降序)，取`NULL`的项是否出现在结果中？如果有，在什么位置?

```
mysql> select * from STC order by score asc;
+-----+-----+-----+-------+
| Sno | Tno | Cno | Score |
+-----+-----+-----+-------+
| S52 | T04 | C07 |  NULL |
| S52 | T05 | C05 |  NULL |
| S02 | T04 | C04 |  NULL |
| S02 | T03 | C03 |  NULL |
| S26 | T07 | C10 |  45.0 |
| S02 | T02 | C02 |  45.0 |
| S52 | T05 | C07 |  53.0 |
| S03 | T01 | C08 |  63.0 |
| S52 | T01 | C08 |  64.0 |
| S02 | T05 | C05 |  70.0 |
| S02 | T01 | C01 |  75.0 |
| S02 | T02 | C09 |  77.0 |
| S03 | T01 | C01 |  78.0 |
| S52 | T02 | C09 |  81.0 |
| S02 | T04 | C06 |  83.0 |
| S02 | T01 | C08 |  83.0 |
| S02 | T07 | C10 |  83.0 |
| S52 | T05 | C03 |  83.0 |
| S01 | T01 | C01 |  83.0 |
| S01 | T03 | C03 |  85.0 |
| S26 | T04 | C04 |  86.0 |
| S02 | T06 | C11 |  88.0 |
| S04 | T06 | C06 |  89.0 |
| S02 | T05 | C07 |  90.0 |
| S52 | T06 | C11 |  90.0 |
| S52 | T07 | C10 |  91.0 |
| S04 | T05 | C05 |  93.0 |
| S03 | T02 | C02 |  93.0 |
+-----+-----+-----+-------+
28 rows in set (0.00 sec)
```

```
mysql> select * from STC order by score desc;
+-----+-----+-----+-------+
| Sno | Tno | Cno | Score |
+-----+-----+-----+-------+
| S04 | T05 | C05 |  93.0 |
| S03 | T02 | C02 |  93.0 |
| S52 | T07 | C10 |  91.0 |
| S52 | T06 | C11 |  90.0 |
| S02 | T05 | C07 |  90.0 |
| S04 | T06 | C06 |  89.0 |
| S02 | T06 | C11 |  88.0 |
| S26 | T04 | C04 |  86.0 |
| S01 | T03 | C03 |  85.0 |
| S02 | T01 | C08 |  83.0 |
| S01 | T01 | C01 |  83.0 |
| S52 | T05 | C03 |  83.0 |
| S02 | T07 | C10 |  83.0 |
| S02 | T04 | C06 |  83.0 |
| S52 | T02 | C09 |  81.0 |
| S03 | T01 | C01 |  78.0 |
| S02 | T02 | C09 |  77.0 |
| S02 | T01 | C01 |  75.0 |
| S02 | T05 | C05 |  70.0 |
| S52 | T01 | C08 |  64.0 |
| S03 | T01 | C08 |  63.0 |
| S52 | T05 | C07 |  53.0 |
| S02 | T02 | C02 |  45.0 |
| S26 | T07 | C10 |  45.0 |
| S02 | T03 | C03 |  NULL |
| S52 | T04 | C07 |  NULL |
| S02 | T04 | C04 |  NULL |
| S52 | T05 | C05 |  NULL |
+-----+-----+-----+-------+
28 rows in set (0.00 sec)
```

> 空值会出现在结果中。
>
> 在`MySQL`中，`NULL`被视作最小值，所以，排升序，它在最上面；排降序，它在最下面。

(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?

```
mysql> select distinct(Score) from STC order by score asc;
+-------+
| Score |
+-------+
|  NULL |
|  45.0 |
|  53.0 |
|  63.0 |
|  64.0 |
|  70.0 |
|  75.0 |
|  77.0 |
|  78.0 |
|  81.0 |
|  83.0 |
|  85.0 |
|  86.0 |
|  88.0 |
|  89.0 |
|  90.0 |
|  91.0 |
|  93.0 |
+-------+
18 rows in set (0.00 sec)
```

```
mysql> select distinct(Score) from STC order by score desc;
+-------+
| Score |
+-------+
|  93.0 |
|  91.0 |
|  90.0 |
|  89.0 |
|  88.0 |
|  86.0 |
|  85.0 |
|  83.0 |
|  81.0 |
|  78.0 |
|  77.0 |
|  75.0 |
|  70.0 |
|  64.0 |
|  63.0 |
|  53.0 |
|  45.0 |
|  NULL |
+-------+
18 rows in set (0.00 sec)
```

> `distinct`有去重的效果，而且`NULL`被视作同一个值。

(5) 通过实验说明使用分组`GROUP BY`对取值为`NULL`的项的处理。

```
mysql> select Score from STC group by Score;
+-------+
| Score |
+-------+
|  83.0 |
|  85.0 |
|  75.0 |
|  45.0 |
|  77.0 |
|  NULL |
|  70.0 |
|  90.0 |
|  88.0 |
|  78.0 |
|  63.0 |
|  93.0 |
|  89.0 |
|  86.0 |
|  64.0 |
|  81.0 |
|  53.0 |
|  91.0 |
+-------+
18 rows in set (0.00 sec)
```

> 所有的`NULL`被视作同一个分组。

(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、最低成绩和总成绩。

```
mysql> select Sno, avg(Score) as 平均成绩, count(*) as 总选课记录数, count(Score) as 总的成绩记录数, max(Score) as 最高成绩, min(Score) as 最低成绩, sum(Score) as 总成
绩 from STC group by Sno;
+-----+-------------+-----------------+---------------------+-------------+-------------+----------+
| Sno | 平均成绩    | 总选课记录数    | 总的成绩记录数      | 最高成绩    | 最低成绩    | 总成绩   |
+-----+-------------+-----------------+---------------------+-------------+-------------+----------+
| S01 |    84.00000 |               2 |                   2 |        85.0 |        83.0 |    168.0 |
| S02 |    77.11111 |              11 |                   9 |        90.0 |        45.0 |    694.0 |
| S03 |    78.00000 |               3 |                   3 |        93.0 |        63.0 |    234.0 |
| S04 |    91.00000 |               2 |                   2 |        93.0 |        89.0 |    182.0 |
| S26 |    65.50000 |               2 |                   2 |        86.0 |        45.0 |    131.0 |
| S52 |    77.00000 |               8 |                   6 |        91.0 |        53.0 |    462.0 |
+-----+-------------+-----------------+---------------------+-------------+-------------+----------+
6 rows in set (0.01 sec)
```

(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。

```
mysql> select count(*) as 总数, avg(Score) as 平均分, max(Score) as 最高分, min(Score) as 最低分 from STC where Score < 0;
+--------+----------+----------+----------+
| 总数   | 平均分   | 最高分   | 最低分   |
+--------+----------+----------+----------+
|      0 |     NULL |     NULL |     NULL |
+--------+----------+----------+----------+
1 row in set (0.00 sec)
```