

数据库实验lab7

曹广杰

15352015 数据科学与计算机

授课教师：刘玉葆

2017/10/20

Content

数据库实验lab7

Content

实验目的

结论汇总

实验内容

Null对元组数目查询的影响

test1

test4

Null在排序中的位置

test2

Null的查询与计算

test3

test5

Null对副词All的影响

test6

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。

结论汇总

1. 在计算元组的算法中，null虽然不能表示明确的值，但是也是会被记为一条记录的。
2. 在升序排列中，被系统当做最小的值处理。
3. 查询的信息就只是非Null的成绩，Null的部分被覆盖了。
4. 在sql的设计中，基本的运算单元为了可以大程度上地表示数据的趋势，都是对Null视而不见的。
5. 对数据进行比较的时候如果有一个比较数据是Null，那么返回值就是空。

实验内容

Null对元组数目查询的影响

test1

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

分析题意：在查询之前，浏览一下当前的问题，发现有多道题目都是关于查询选修课程C++，为了之后的方便，先为c++的选课信息创建一个view，以便后续使用：

```
create view CPP as
select *
from CHOICES
where CHOICES.cid=
    (select cid
     from COURSES
     where COURSES.cname = 'c++')
```

之后就是基于这个view的查询操作了——

查询选课人数：

```
select COUNT(*)
from CHOICES
where cid = '10005'
```

查询成绩及格人数：

```
select COUNT(*) as accept
from cpp
where score >= 60
```

查询成绩不及格人数：

```
select COUNT(*) as wans
from cpp
where score < 60
```

查询成绩为null的人数：

```
select COUNT(*) as nonexist
from cpp
where score is null
```

综上：

这里的查询结果表明，在计算元组的算法中，null虽然不能表示明确的值，但是也是会被记为一条记录的。

test4

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？

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

在查询grade信息的时候，查柜情况下是不会出现类似于未知年级的情况的。但是由于数据库中的数据存储等等原因，在查询中有Null的存在。

Null在排序中的位置

test2

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 cpp
order by score asc
```

根据查询结果得知：

查询的结果是每一个学号对应一个c++课程的成绩，这里Null是存在的，在升序排列中，Null在最上面，被系统当做最小的值处理。

Null的查询与计算

test3

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 cpp
where score is null
order by score asc
```

题意分析：

本次查询中，sid与score二者呈对应关系。那么在保留字“distinct”作用在哪里呢？

- 如果保留字作用在score上，就意味着一个score只能出现一次，那么有多个学号有同样的分数，学号也就只能出现一次——这就造成了信息的丢失。
- 如果作用在保留字上，一个学生可能有多个成绩——但是由笔者实验得知，在本次实验中，每位学生至多有两个成绩，而且有两个成绩的学号中一定有一个成绩显示为Null。

为了保持程序的正常运行，如果我是设计者，我会选择使用第二种方案，事实上sql的设计也是这样——distinct只能添加在sid的前面作为限定。

这里就涉及到Null与已有成绩的取舍问题。在一个人有两个成绩二其中一个是Null的时候，我们在查询中希望见到哪一个成绩呢？很显然，Null的成绩是没有意义的，故而在本次查询中，查询的信息就只是非Null的成绩，Null的部分被覆盖了。

test5

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。

```
select AVG(score) as average,
       COUNT(*) as cnt,
       MAX(score) as maxone,
       MIN(score) as minone
from CHOICES
group by cid
```

查询显示以上四种查询的信息都没有出现Null的信息，这就意味着在求和运算中，Null从来都没有被纳入考虑中。

综上所述，在sql的设计中，基本的运算单元为了可以大程度上地表示数据的趋势，都是对Null视而不见的，这样就避免了平时数据中经常出现的Null对于数据分析的不利影响。

Null对副词All的影响

test6

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。

```
select grade
from STUDENTS as new
where new.grade
      >= all
      (select grade
       from STUDENTS as whole
       where whole.grade is not null)
```

之前说过在排序中，Null总是处于靠前的位置，这似乎在暗示我们Null已经被默认为最小值。

但是在嵌套查询语句中，谓词All的出现却受到了Null更为深刻的影响。

All表示之前的new.grade应该满足whole中的所有数据对应的条件——如果这个条件副词设置为any或者some，那么副词后的查询语句是可以给出一个精准的答案的，但是对于all来说，这是不可行的，all需要对所有的数据都进行比较——这里如果有一个比较数据是Null，那么返回值就是Null：

```
select grade
from STUDENTS as new
where new.grade
      >= null
```

在这条查询语句中，查询的结果是空，而不是将Null作为最小值来判断的。这种设计是基于Null并不是一个真实的数字，而是与真实的数字有所不同的，如果将其当做真实的数字进行处理，那么之前的Null的查询覆盖就会影响到真实数字的查询——一个人有60分和68分，则只会查询到68分，较小的数字会如同Null一样被覆盖，这显然是不合逻辑的。