

19335262 张航悦 实验 8

19335262 张航悦 实验 8

- 练习1
- 练习2
- 练习3
- 练习4
- 练习5
- 练习6

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。

- 查询选修课程C++的学生的总人数，为6031人

```
SQLQuery1.sql -...M6B5L\18230 (54))*  X
```

```
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
```

100 %

结果 消息

	(无列名)
1	6031

- 查询成绩合格的人数，为4817人

```
SQLQuery1.sql -...M6B5L\18230 (54))*  X
```

```
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
and CHOICES.score>=60
```

100 %

结果 消息

	(无列名)
1	4817

- 查询成绩不合格的人数，为724人

```
SQLQuery1.sql -...M6B5L\18230 (54))*  X
```

```
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
and CHOICES.score<60
```

100 %

结果 消息

	(无列名)
1	724

- 讨论剩余的 $6031 - 4817 - 724 = 490$ 人的情况。由于当score取NULL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490个score为NULL的元组。以

下为代码验证。

SQLQuery1.sql -...M6B5L\18230 (54))*	
<pre>from CHOICES, COURSES where CHOICES.cid=COURSES.cid and COURSES.cname='C++' and CHOICES.score is null</pre>	
100 %	
结果	消息
(无列名)	
1	490

练习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？

- 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。

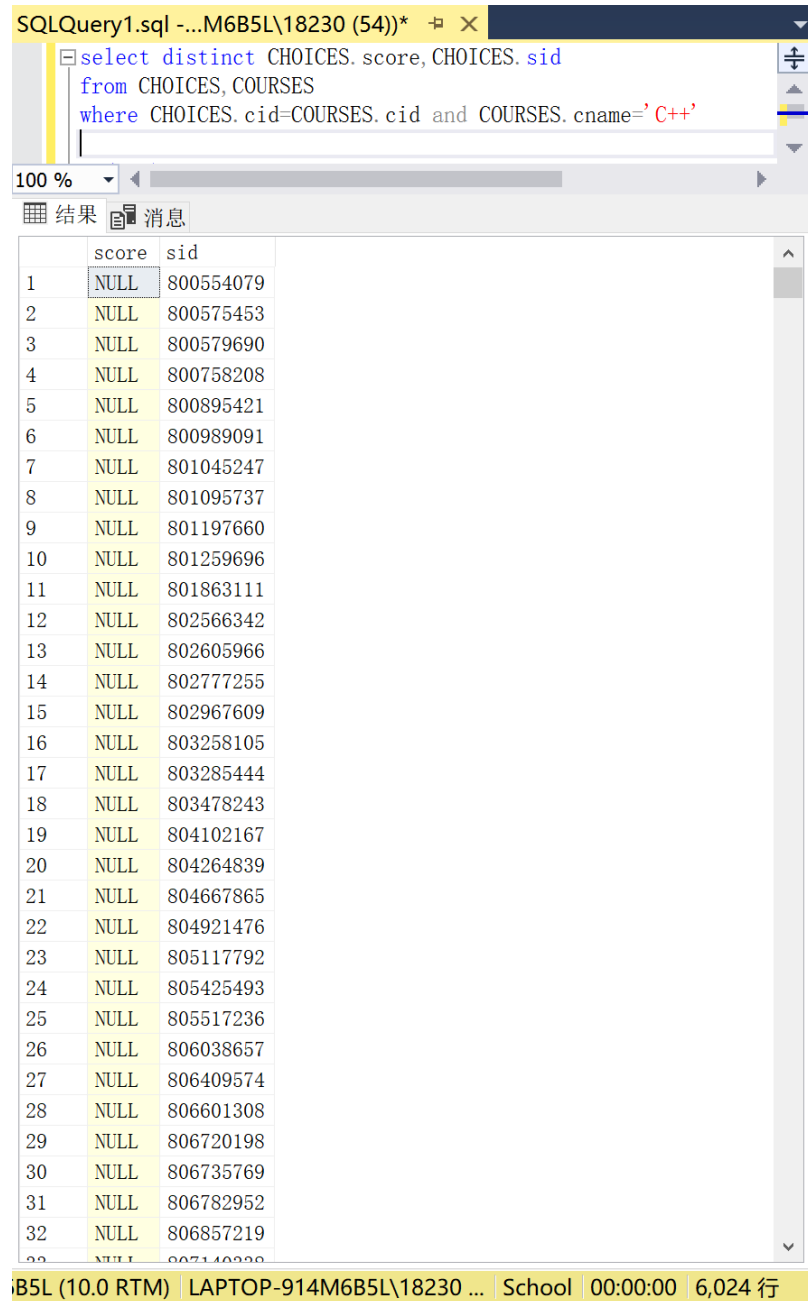
SQLQuery1.sql -...M6B5L\18230 (54))*	
<pre>select CHOICES.sid, CHOICES.score from CHOICES, COURSES where CHOICES.cid=COURSES.cid and COURSES.cname='C++' order by CHOICES.score asc</pre>	
100 %	
结果	消息
sid	score
1 845947855	NULL
2 890918686	NULL
3 898137922	NULL
4 867715893	NULL
5 872519782	NULL
6 886109186	NULL
7 884877167	NULL
8 893869264	NULL
9 834918408	NULL
10 873557912	NULL
11 802605966	NULL
12 878863812	NULL
13 812437303	NULL
14 824027469	NULL
15 849813323	NULL
16 878672282	NULL
17 858221388	NULL
18 894891572	NULL
19 822628145	NULL
20 866193870	NULL
21 882446872	NULL
22 811059947	NULL
23 822113495	NULL
24 805517236	NULL
25 878539525	NULL
26 839269508	NULL
27 871236570	NULL
28 817940769	NULL
29 845698872	NULL
30 892443563	NULL
31 840795078	NULL
32 884618909	NULL
33 871588492	NULL

B5L (10.0 RTM) LAPTOP-914M6B5L\18230 ... School 00:00:00 6,031 行

练习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？

- 使用保留字DISTINCT时，会同时作用在 score 和 sid 这两个字段上，因此只有 score 和 sid 都相同的元组才会被删除。



The screenshot shows a SQL query window titled "SQLQuery1.sql -...M6B5L\18230 (54))*". The query is:

```
select distinct CHOICES. score, CHOICES. sid
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='C++'
```

The results are displayed in a table with columns "score" and "sid". The "score" column contains 32 NULL values, and the "sid" column contains 32 unique numeric values. The status bar at the bottom indicates "6,024 行" (6,024 rows).

	score	sid
1	NULL	800554079
2	NULL	800575453
3	NULL	800579690
4	NULL	800758208
5	NULL	800895421
6	NULL	800989091
7	NULL	801045247
8	NULL	801095737
9	NULL	801197660
10	NULL	801259696
11	NULL	801863111
12	NULL	802566342
13	NULL	802605966
14	NULL	802777255
15	NULL	802967609
16	NULL	803258105
17	NULL	803285444
18	NULL	803478243
19	NULL	804102167
20	NULL	804264839
21	NULL	804667865
22	NULL	804921476
23	NULL	805117792
24	NULL	805425493
25	NULL	805517236
26	NULL	806038657
27	NULL	806409574
28	NULL	806601308
29	NULL	806720198
30	NULL	806735769
31	NULL	806782952
32	NULL	806857219

练习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？

将得到15个组。发现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。

SQLQuery1.sql - ...M6B5L\18230 (54))*	
<pre> select grade from STUDENTS group by grade </pre>	
100 %	
<div> <div>结果</div> <div>消息</div> </div>	
	grade
1	1998
2	2004
3	2001
4	1996
5	2002
6	1999
7	1993
8	2000
9	1994
10	1997
11	1991
12	NULL
13	2003
14	1992
15	1995

练习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。

可以发现使用count(*)时得到的count1和使用count(score)时得到的count2并不相同。当记录为NULL时,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。

虽然记录中存在NULL,但课程最低成绩仍非NULL说明聚集函数 `min()` 会忽略NULL值。同时,聚集函数 `sum()`, `max()`, `avg()` 均会忽略NULL值处理。

SQLQuery1.sql -...M6B5L\18230 (54))*

```

select cid, avg(score) as avg, count(*) as count1,
count(score) as count2, max(score) as maxscore, min(score) as minscore
from CHOICES
group by cid

```

100 %

结果 消息

	cid	avg	count1	count2	maxscore	minscore
1	10008	75	5985	5512	99	50
2	10019	75	6074	5555	99	50
3	10018	76	5969	5520	99	50
4	10040	75	6102	5596	99	50
5	10011	76	6086	5621	99	50
6	10028	75	6042	5567	99	50
7	10035	76	6104	5607	99	50
8	10021	76	5916	5462	99	50
9	10046	76	6069	5606	99	50
10	10032	75	6073	5614	99	50
11	10005	76	6031	5541	99	50
12	10050	75	6077	5609	99	50
13	10034	76	6000	5506	99	50
14	10015	75	5990	5508	99	50
15	10036	75	6043	5547	99	50
16	10045	76	6041	5549	99	50
17	10025	75	5955	5478	99	50
18	10030	76	5985	5486	99	50
19	10049	76	5945	5474	99	50
20	10007	75	5965	5494	99	50
21	10026	75	5881	5402	99	50
22	10010	75	6028	5517	99	50
23	10024	75	6013	5554	99	50
24	10033	75	6039	5558	99	50
25	10041	76	6027	5547	99	50
26	10002	75	6012	5501	99	50
27	10020	76	6027	5567	99	50
28	10037	75	5909	5457	103	50
29	10048	76	5926	5417	99	50
30	10016	76	5867	5420	99	50
31	10039	76	5951	5502	99	50
32	10004	76	6110	5639	99	50

6B5L (10.0 RTM) LAPTOP-914M6B5L\18230 ... School 00:00:00 50 行

练习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

发现查询结果返回为空。

SQLQuery1.sql -...M6B5L\18230 (54))*

```

select grade
from STUDENTS as s1
where grade >= all (select grade
from STUDENTS as s2
where s1.sid != s2.sid)

```

100 %

结果 消息

grade

查询students表中所有的年级可能取值发现有NULL值。对NULL值进行任何比较运算，由于均无法判断是真是假，将返回unknown，所以不会有任何输出。

SQLQuery2.sql -...M6B5L\18230 (53))*

```
select distinct grade
from STUDENTS
```

100 %

结果 消息

	grade
1	1998
2	2004
3	2001
4	1996
5	2002
6	1999
7	1993
8	2000
9	1994
10	1997
11	1991
12	NULL
13	2003
14	1992
15	1995