

# Database-System Experiment-9

---

21307289 刘森元

## 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。

## 2. 实验环境

---

Macbook Pro 2021 (Apple M1 Pro)

macOS Ventura 13.5.2

PostgreSQL 15.4 (Homebrew)

zsh 5.9

## 3. 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。

## 4. 实验步骤

---

完整代码见附件 [Experiment-9.sql](#)

使用命令 `psql --dbname=postgres --file=Experiment-9.sql --echo-errors --quiet` 以运行脚本。

## 重建数据库

```
-- 重建数据库
DROP DATABASE "school";
DROP USER USER1;
DROP USER USER2;
DROP USER USER3;
CREATE DATABASE "school";
\c school
\cd '/Users/qiu_nangong/Documents/Github/Database-System/Experiment-8'
\i STUDENTS.sql
\i TEACHERS.sql
\i COURSES.sql
\i CHOICES.sql
```

## 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。

```
SELECT no, sid, cid, tid,
       CASE
         WHEN score IS NULL THEN NULL
         ELSE ROUND(score / 20.0, 3)
       END AS converted_score
FROM CHOICES;
```

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。

```
SELECT COUNT(*) AS total_students,
       COUNT(CASE WHEN score >= 60 THEN 1 END) AS passed_students,
       COUNT(CASE WHEN score < 60 THEN 1 END) AS failed_students
FROM CHOICES
WHERE cid = '10028';
```

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？

```
SELECT *  
FROM choices  
ORDER BY score DESC;
```

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？

在前面的查询中加入DISTINCT关键字将从结果中删除重复的行，确保每行基于所选列的唯一性。

5) 通过实验说明使用分组GROUP BY对取值为NULL的项的处理。

在前面的查询中加入DISTINCT关键字将从结果中删除重复的行，确保每行基于所选列的唯一性。

使用分组GROUP BY对取值为NULL的项的处理方式如下：

首先，我们创建一个示例表格来演示：

```
CREATE TABLE example (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50),  
    score INT  
);  
INSERT INTO example (name, score) VALUES  
    ('Alice', 80),  
    ('Bob', NULL),  
    ('Charlie', 70),  
    ('Dave', NULL),  
    ('Eve', 90);
```

现在，我们将对该表格使用分组GROUP BY，并观察对NULL值的处理方式：

```
SELECT name, AVG(score) AS average_score  
FROM example  
GROUP BY name;
```

结果如下：

| name    | average_score |
|---------|---------------|
| Alice   | 80            |
| Bob     |               |
| Charlie | 70            |
| Dave    |               |
| Eve     | 90            |

在分组GROUP BY的结果中，对于NULL值，它们会被作为一个独立的分组显示，并且在聚合函数中显示为空值。

在上述示例中，Bob和Dave的分数为NULL，因此它们各自被作为一个独立的分组显示，并在average\_score列中显示为空值。

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。

```
SELECT sid, AVG(COALESCE(score, 0)) AS average_score,
       COUNT(*) AS total_records,
       MAX(score) AS max_score,
       MIN(score) AS min_score,
       SUM(CASE WHEN score IS NULL THEN 0 ELSE score END) AS total_score
FROM choices
GROUP BY sid;
```

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。

```
SELECT COUNT(*) AS total_records,
       AVG(score) AS average_score,
       MAX(score) AS max_score,
       MIN(score) AS min_score
FROM choices
WHERE score < 0;
```

## 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。

```
SELECT cid, COALESCE(SUM(hour), 0) AS hour, COALESCE(SUM(hour), 0) * 18 AS
total_hour
FROM courses
GROUP BY cid;
```

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。

```
SELECT COUNT(DISTINCT sid) AS total_students,
       COUNT(DISTINCT CASE WHEN score >= 60 THEN sid END) AS passed_students,
       COUNT(DISTINCT CASE WHEN score < 60 THEN sid END) AS failed_students
FROM CHOICES
WHERE cid = 'c++';
```

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?

```
SELECT sid
FROM CHOICES
ORDER BY score;
```

出现在升序排序的表末

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?

如果在查询中加入DISTINCT关键字，它将从结果集中消除重复的行

5) 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 COUNT(DISTINCT grade) AS group_number
FROM STUDENTS;
```

与现实情况相比，查询结果给出了按年级分组的数量。然而，需要注意的是，查询结果可能与实际情况不完全一致。实际情况中，可能存在一些特殊情况，例如有些学生可能没有年级，或者年级的定义可能不仅仅是一个简单的数值。因此，实际的学生分组数量可能会有所不同。

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。

```
SELECT cid,  
       AVG(score) AS average_score,  
       COUNT(*) AS total_records,  
       MAX(score) AS max_score,  
       MIN(score) AS min_score  
FROM CHOICES  
GROUP BY cid;
```

关于空值对集合函数的影响，集合函数通常会忽略空值（NULL），除非特别指定。在这个查询中，如果某门课程的选修学生中存在空值（NULL），集合函数将忽略这些空值并计算其他非空值的结果。因此，空值对集合函数的作用是被忽略，不会影响计算结果。

## 5. 实验心得

---

通过本次实验，我对数据库中的NULL值和空集有了更深入的理解。我能够熟练地使用SQL语句来处理和操作这些特殊值，从而更好地处理和分析数据库中的数据。这对于正确处理和解释数据非常重要，尤其是在数据分析和决策过程中。