

Problem

Submissions

Leaderboard

Discussions

Pivot the Occupation column in **OCCUPATIONS** so that each Name is sorted alphabetically and displayed underneath its corresponding Occupation. The output column headers should be Doctor, Professor, Singer, and Actor, respectively.

**Note:** Print **NULL** when there are no more names corresponding to an occupation.

**Input Format**

The **OCCUPATIONS** table is described as follows:

Column	Type
Name	String
Occupation	String

Occupation will only contain one of the following values: **Doctor, Professor, Singer** or **Actor**.

**Sample Input**

Name	Occupation
Samantha	Doctor
Julia	Actor
Maria	Actor
Meera	Singer
Ashely	Professor
Ketty	Professor

Current Buffer (saved locally, editable)

MySQL

```
1 SET @r1=0, @r2=0, @r3 =0, @r4=0;
2 SELECT MIN(Doctor), MIN(Professor), MIN(Singer), MIN(Actor) FROM
3 (SELECT CASE Occupation WHEN 'Doctor' THEN @r1:=@r1+1
4           WHEN 'Professor' THEN @r2:=@r2+1
5           WHEN 'Singer' THEN @r3:=@r3+1
6           WHEN 'Actor' THEN @r4:=@r4+1 END
7      AS RowLine,
8      CASE WHEN Occupation = 'Doctor' THEN Name END AS Doctor,
9      CASE WHEN Occupation = 'Professor' THEN Name END AS Professor,
10     CASE WHEN Occupation = 'Singer' THEN Name END AS Singer,
11     CASE WHEN Occupation = 'Actor' THEN Name END AS Actor
12     FROM OCCUPATIONS ORDER BY Name) AS t
13 GROUP BY RowLine;
```

Line: 1 Col: 1

⬆ Upload Code as File

Run Code

Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✔ Sample Test case 0

Your Output (stdout)

```
1 Aamina Ashley Christeen Eve
2 Julia Belvet Jane Jennifer
3 Priya Britney Jenny Ketty
4 NULL Maria Kristeen Samantha
5 NULL Meera NULL NULL
6 NULL Naomi NULL NULL
7 NULL Priyanka NULL NULL
```

Problem

Pivot the Occupation column in **OCCUPATIONS** so that each Name is sorted alphabetically and displayed underneath its corresponding Occupation. The output column headers should be Doctor, Professor, Singer, and Actor, respectively.

**Note:** Print **NULL** when there are no more names corresponding to an occupation.

Input Format

The **OCCUPATIONS** table is described as follows:

Submissions

Leaderboard

Discussions

Column	Type
Name	String
Occupation	String

Occupation will only contain one of the following values: **Doctor**, **Professor**, **Singer** or **Actor**.

Sample Input

Name	Occupation
Samantha	Doctor
Julia	Actor
Maria	Actor
Meera	Singer
Ashely	Professor
Ketty	Professor

Current Buffer (saved locally, editable)

MySQL

```
1 SET @r1=0, @r2=0, @r3 =0, @r4=0;
2 SELECT MIN(Doctor), MIN(Professor), MIN(Singer), MIN(Actor) FROM
3 (SELECT CASE Occupation WHEN 'Doctor' THEN @r1:=@r1+1
4           WHEN 'Professor' THEN @r2:=@r2+1
5           WHEN 'Singer' THEN @r3:=@r3+1
6           WHEN 'Actor' THEN @r4:=@r4+1 END
7      AS RowLine,
8      CASE WHEN Occupation = 'Doctor' THEN Name END AS Doctor,
9      CASE WHEN Occupation = 'Professor' THEN Name END AS Professor,
10     CASE WHEN Occupation = 'Singer' THEN Name END AS Singer,
11     CASE WHEN Occupation = 'Actor' THEN Name END AS Actor
12     FROM OCCUPATIONS ORDER BY Name) AS t
13 GROUP BY RowLine;
```

Line: 1 Col: 1

Upload Code as File

Run Code

Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Your Output (stdout)

```
1 Aamina Ashley Christeen Eve
2 Julia Belvet Jane Jennifer
3 Priya Britney Jenny Ketty
4 NULL Maria Kristeen Samantha
5 NULL Meera NULL NULL
6 NULL Naomi NULL NULL
7 NULL Priyanka NULL NULL
```