University of Toronto
CSC343, Winter 2017

GROUP BY and HAVING: Solutions

Schema

Student(sID, surName, firstName, campus, email, cgpa)

Course(dept, cNum, name, breadth)

Offering(oID, dept, cNum, term, instructor)

Took(sID, oID, grade)

Offering[dept, cNum] $\subseteq$ Course[dept, cNum]

Took[sID] $\subseteq$ Student[sID]

Took[oID] $\subseteq$ Offering[oID]

Questions

1. Write a query to find the average grade, minimum grade, and maximum grade for each offering.

   Solution:

   ```
   select avg(grade), min(grade), max(grade)
   from Took
   group by oid;
   ```

   Output:

   ```
            avg          | min | max
   ----------------------+-----+-----
    59.0000000000000000 |  39 |  98
    60.6666666666666667 |  45 |  75
    70.5000000000000000 |  52 |  89
   . . . rows omitted
    75.0000000000000000 |  54 |  96
    78.0000000000000000 |  78 |  78
    83.0000000000000000 |  71 |  89
   (23 rows)
   (1 row)
   ```

2. Suppose we wrote

   ```
   SELECT ________________________
   FROM Offering, Took
   WHERE Offering.oID = Took.oID
   group by dept;
   ```

   Which of the following could go in the SELECT clause?

   `sID` `count(sID)` `grade` `avg(grade)` `dept` `count(dept)` `term` `min(term)`

   Solution: The only unaggregated item that can go in the SELECT is the one that is grouped by: `dept`. Everything else must be aggregated. And it is legal to aggregate `dept` too. Here is a query with all the allowed items from our list:

```
SELECT count(sID), avg(grade), dept, count(dept), min(term)
FROM Offering, Took
WHERE Offering.oID = Took.oID
group by dept;
```

It makes sense that these are allowed. We asked to group by dept, so there will be one row per dept in the resulting table. This means that we can include in our SELECT only things that have one value per per dept. There is one count(sID) per dept, one avg(grade) per dept, one dept per dept (so to speak), one count(dept) per dept, and one min(term) per department.

Output:

```
 count |         avg         | dept | count |  min
-------+---------------------+------+-------+-------
     4 | 69.5000000000000000 | ENV  |     4 | 20089
     6 | 78.1666666666666667 | EEB  |     6 | 20081
     8 | 78.5000000000000000 | ANT  |     8 | 20081
     1 | 97.0000000000000000 | HIS  |     1 | 20081
    24 | 79.6666666666666667 | CSC  |    24 | 20081
    11 | 63.6363636363636364 | ENG  |    11 | 20081
(6 rows)
```

3. Find the sid and minimum grade of each student with an average over 80.

Solution:

```
SELECT SID, min(grade)
FROM Took
GROUP BY sID
HAVING AVG(grade) > 80;
```

Output:

```
  sid  | min
-------+-----
 98000 |  54
 99999 |  52
(2 rows)
```

4. Find the sid, surname, and average grade of each student, but keep the data only for those students who have taken at least 3 courses.

   Solution:

```
SELECT Student.sID, surname, avg(grade)
FROM Student, Took
WHERE Student.sID = Took.sID
GROUP BY Student.sID
HAVING count(grade) >= 10;
```

   Output:

```
  sid  |  surname   |         avg
-------+------------+---------------------
 98000 | Fairgrieve | 83.2000000000000000
 99999 | Ali        | 84.5833333333333333
   157 | Lakemeyer  | 75.9333333333333333
(3 rows)
```

5. For each student who has passed at least 10 courses, report their sid and average grade on the courses that they passed.

   Solution:

```
SELECT sid, AVG(grade), COUNT(*)
FROM took
WHERE grade >= 50
GROUP BY sid
HAVING count(*) >= 10;
```

   Output:

```
  sid  |         avg          | count
-------+----------------------+-------
 98000 | 83.2000000000000000 |    15
 99999 | 84.5833333333333333 |    12
   157 | 78.5714285714285714 |    14
(3 rows)
```

   There is a lot going on here. Be sure you are clear on the difference between WHERE and HAVING, and which rows are left at the moment where the HAVING condition is checked for each group.

6. Which of these queries is legal?

```
SELECT dept
FROM Took, Offering
WHERE Took.oID = Offering.oID
GROUP BY dept
HAVING avg(grade) > 75;
```

```
SELECT Took.oID, avg(grade)
FROM Took, Offering
WHERE Took.oID = Offering.oID
GROUP BY Took.oID
HAVING avg(grade) > 75;
```

```
SELECT Took.oID, dept, cNum, avg(grade)
FROM Took, Offering
WHERE Took.oID = Offering.oID
GROUP BY Took.oID
HAVING avg(grade) > 75;
```

```
SELECT oID, avg(grade)
FROM Took
GROUP BY sID
HAVING avg(grade) > 75;
```

Solution: Here's the result of each:

```
 dept
------
 EEB
 ANT
 HIS
 CSC
(4 rows)
```

```
 oid |         avg
-----+---------------------
   8 | 92.0000000000000000
  28 | 91.0000000000000000
. . .  rows omitted
   7 | 83.0000000000000000
(11 rows)
```

```
ERROR:  column "offering.dept" must appear
in the GROUP BY clause or be used in an
aggregate function
LINE 1: SELECT Took.oID, dept,
    cNum, avg(grade)
```

```
ERROR:  column "took.oid" must appear
in the GROUP BY clause or be used in an
aggregate function
LINE 1: SELECT oID, avg(grade)
```