University of Toronto
CSC343

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. Which of these queries is legal?

   ```
   SELECT surname, sid
   FROM Student, Took
   WHERE Student.sid = Took.sid
   GROUP BY sid;
   ```

   ```
   SELECT surname, Student.sid
   FROM Student, Took
   WHERE Student.sid = Took.sid
   GROUP BY campus;
   ```

   ```
   SELECT instructor, max(grade),
       count(Took.oid)
   FROM Took, Offering
   WHERE Took.oid = Offering.oid
   GROUP BY instructor;
   ```

   ```
   SELECT Course.dept, Course.cnum,
       count(oid), count(instructor)
   FROM Course, Offering
   WHERE Course.dept = Offering.dept and
       Course.cnum = Offering.cnum
   GROUP BY Course.dept, Course.cnum
   ORDER BY count(oid);
   ```

Solution: Here's the result of each:

```
SELECT surname, sid
FROM Student, Took
WHERE Student.sid = Took.sid
GROUP BY sid;
ERROR:  column reference "sid" is ambiguous
LINE 1: SELECT surname, sid
                        ^
```

```
 instructor | max | count
------------+-----+-------
 Heap       |  82 |     1
 Miller     |  91 |     1
 Johancsik  |  99 |     3
. . . etc.
 Mylopoulos |  96 |     3
 Percy      |  98 |     4
 Mendel     |  75 |     3
(17 rows)
```

```
ERROR:  column "student.surname" must
appear in the GROUP BY clause or be used in
an aggregate function
LINE 1: SELECT surname, Student.sid
               ^
```

```
 dept | cnum | count | count
------+------+-------+-------
 ENV  |  200 |     1 |     1
. . . etc.
 CSC  |  263 |     3 |     3
 CSC  |  148 |     4 |     4
 CSC  |  207 |     4 |     4
 CSC  |  343 |     5 |     5
(17 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 10 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)
FROM took
WHERE grade >= 50
GROUP BY sid
HAVING count(*) >= 10;
```

   Output:

```
  sid  |         avg
-------+---------------------
 98000 | 83.2000000000000000
 99999 | 84.5833333333333333
   157 | 78.5714285714285714
(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. For each student who has passed at least 10 courses, report their sid and average grade on *all of their courses*.

   Solution: Here, because we don't want a filter applied (only passing grades count) when choosing which students to report on, but we don't want that filter applied when we compute their average grade. A single query, with a single WHERE clause, can't accomplish this. Views to the rescue!

```
CREATE VIEW Seniors AS
SELECT sid
FROM Took
WHERE grade >= 50
GROUP BY sid
HAVING count(*) >= 10;
```

```
SELECT Seniors.sid, AVG(grade)
FROM Seniors, Took
WHERE seniors.sid = Took.sid
GROUP BY Seniors.sid;
```

Output:

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

Notice that the average for student 157 is different than it was in the previous question. This is because that student failed one course, and it now is allowed to pull down the reported average.

7. 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)
```