University of Toronto
CSC343, Fall 2015

CDF IDs:
Names:

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 included:

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

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 average grade of each student, but keep the data only for those students who have an sid over 22222.

   Solution:

```
SELECT Student.sID, surname, avg(grade)
FROM Student, Took
WHERE Student.sID = Took.sID
GROUP BY Student.sID
HAVING Student.sID > 22222;
```

   Output:

```
  sid  |  surname   |         avg
-------+------------+---------------------
 98000 | Fairgrieve | 83.2000000000000000
 99132 | Marchmount | 76.2857142857142857
 99999 | Ali        | 84.5833333333333333
(3 rows)
```

4. Find only the sid (and not also the average grade) of each student with an average over 80.

   Solution:

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

   Output:

```
  sid
-------
 98000
 99999
(2 rows)
```

5. 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)
```