University of Toronto
CSC343, Summer 2019

Set Operations: 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. Answer each of the following questions with an arithmetic expression.

   Suppose a row occurs n times in table R and m times in table S.

   (a) Using bag semantics, how many times will it occur in table $R \cup S$?
   (b) Using bag semantics, how many times will it occur in table $R \cap S$?
   (c) Using bag semantics, how many times will it occur in table $R - S$?

   Solution:

   (a) $n + m$
   (b) $min(n,\ m)$
   (c) $max(n - m,\ 0)$

2. Use a set operation to find all terms when Jepson and Suzuki were both teaching. Include every occurrence of a term from the result of both operands.

   Solution:

```
(select Term from Offering where instructor = ’Suzuki’)
intersect all
(select Term from Offering where instructor = ’Jepson’);
```

   Output:

```
 term
-------
 20089
 20081
 20081
(3 rows)
```

3. Find the sID of students who have earned a grade of 85 or more in some course, or who have passed a course taught by Atwood. Ensure that no sID occurs twice in the result.

   Solution:

```
create view High as
(select sid from took where grade >= 85);
```

```
create view HighAtwood as
(select sid
from Took, Offering
where Took.oid = Offering.oid and instructor = 'Atwood' and grade >= 50);

(select * from high)
union
(select * from highAtwood);
```

Output:

```
  sid
-------
 98000
 99132
 99999
   157
(4 rows)
```

4. Find all terms when csc369 was not offered.

   Solution:

```
(select term
from Offering)
except
(select term
from Offering
where dept = 'csc' and cNum = 369);
```

Output:

```
 term
-------
 20081
 20089
```

5. Make a table with two columns: oID and results. In the results column, report either “high” (if that offering had an average grade of 80 or higher), or “low” (if that offering had an average under 60). Offerings with an average in between will not be included.

   Hints:

   - Surprise surprise, use a set operation.
   - You can use the SELECT clause to put a literal value into a column. For example:
     `SELECT 'high' as results ....`

   Solution:

```
(SELECT oID, 'high' AS results
FROM Took
GROUP BY oID
HAVING avg(grade) >=80)
UNION
(SELECT oID, 'low' as results
FROM Took
GROUP by oID
HAVING avg(grade) < 60);
```

   Output:

```
 oid | results
-----+---------
  39 | high
   3 | high
  14 | low
  38 | high
   7 | high
   8 | high
  28 | high
   1 | high
  15 | low
  13 | high
(10 rows)
```