Chun-Wei

- Home Page
- Assignments Due
- Progress Report
- Handouts
- Tutorials
- Homeworks
- Lab Projects
- Log Out

Help

Gradiance Online Accelerated Learning

Submission number:	145169
Submission certificate:	CE559661
Submission time:	2014-01-16 15:34:17 PST (GMT - 8:00)

Number of questions:	9
Positive points per question:	3.0
Negative points per question:	1.0
Your score:	27

1. The table

Scores(Team, Day, Opponent, Runs)

Gives the scores in the Japanese Baseball League for two consecutive days. The Opponent is NULL if the Team did not play on that day. The number of Runs is given as NULL if either the team did not play, or will play on that day but the game is not yet concluded. The data in this table is as follows:

Team	Day	Opponent	Runs
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	NULL	NULL
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	NULL	NULL
Dragons	Monday	Carp	NULL
Tigers	Monday	NULL	NULL
Carp	Monday	Dragons	NULL
Swallows	Monday	Giants	0
Bay Stars	Monday	NULL	NULL
Giants	Monday	Swallows	5

What is the result of the following query?

```
SELECT Team, Min(Opponent), Max(Runs)
FROM Scores
GROUP BY Team
```

Note: When a column has a string type, Min and Max refer to lexicographic (alphabetic) order of strings. That is, Min(Opponent) is the opponent that comes first in alphabetical order, and Max(Opponent) would be the one that comes last in alphabetical order.

Identify in the list below one of the tuples in the result.

a)

Dragons	Swallows	4

b)

Bay Stars	NULL	2

c)

Bay Stars	Tigers	2

d)

Tigers	NULL	9

Answer submitted: **c)**

You have answered the question correctly.

Question Explanation:

Remember that NULL does not count in any aggregation, so it doesn't matter whether "NULL" precedes or follows an opponent name alphabetically. Also, if there is nothing to take an aggregation over, as is the case for the Max(Runs) in the group for the Carp, then the result is NULL. Here is the resulting table:

Name	MIN(Opponent)	Max(Runs)
Dragons	Carp	4
Tigers	Bay Stars	9
Carp	Dragons	NULL
Swallows	Dragons	7
Bay Stars	Tigers	2
Giants	Swallows	5

2. Suppose we have a relation with schema

```
R(A, B, C, D, E)
```

If we issue a query of the form

```
SELECT ...
FROM R
WHERE ...
GROUP BY B, E
HAVING ???
```

What terms can appear in the HAVING condition (represented by ??? in the above query)? Identify, in the list below, the term that CAN NOT appear.

a) A

b) AVG(A)

c) SUM(D)

d) COUNT(*)

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Any aggregated term can appear in the HAVING condition. However, an attribute that is not in the GROUP-BY list cannot appear, unaggregated, in the HAVING condition. Thus, B or E may appear unaggregated, and all five attributes can appear in an aggregation. However, A, C, or D cannot appear by themselves.

3. The table

Scores(Team, Day, Opponent, Runs)

Gives the scores in the Japanese Baseball League for two consecutive days. The data in this table is as follows:

Team	Day	Opponent	Runs
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	Giants	2
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	Carp	4
Dragons	Monday	Carp	6
Tigers	Monday	Bay Stars	5
Carp	Monday	Dragons	3
Swallows	Monday	Giants	0
Bay Stars	Monday	Tigers	7
Giants	Monday	Swallows	5

Determine the result of the query

```
SELECT Team, SUM(Runs)
FROM Scores
GROUP BY Team
```

Then, identify, in the list below, one of the tuples of the result.

a)

Bay Stars	9

b)

Swallows	0

c)

Carp	54

d)

Giants	34

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Each team played two games, so the team appears in the result next to the sum of the numbers of runs they scored in those two games. The result is:

Team	SUM(Runs)
Dragons	10
Tigers	14
Carp	5
Swallows	7
Bay Stars	9
Giants	9

4. Suppose we have a relation with schema

```
R(A, B, C, D, E)
```

If we issue a query of the form

```
SELECT ???
FROM R
WHERE ...
GROUP BY C, D
```

What terms can appear in the SELECT list (represented by ??? in the above query)? Identify, in the list below, the term that CAN NOT appear.

a) D

b) C

c) SUM(E)

d) E

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

When there is a GROUP-BY clause, any aggregated term can appear in the SELECT list. However, an attribute that is not in the GROUP-BY list cannot appear, unaggregated, in the SELECT list. Thus, C or D may appear unaggregated, and all five attributes can appear in an aggregation. However, A, B, or E cannot appear by themselves.

5. The table

Scores(Team, Day, Opponent, Runs)

Gives the scores in the Japanese Baseball League for two consecutive days. The Opponent is NULL if the Team did not play on that day. The number of Runs is given as NULL if either the team did not play, or will play on that day but the game is not yet concluded. The data in this table is as follows:

Team	Day	Opponent	Runs
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	NULL	NULL
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	NULL	NULL
Dragons	Monday	Carp	NULL
Tigers	Monday	NULL	NULL
Carp	Monday	Dragons	NULL
Swallows	Monday	Giants	0
Bay Stars	Monday	NULL	NULL
Giants	Monday	Swallows	5

What is the result of the following query?

```
SELECT Opponent, COUNT(*), AVG(Runs)
FROM Scores
GROUP BY Opponent
```

Then, identify in the list below one of the tuples in the result.

a)

Bay Stars	12	2.667

b)

Carp	1	NULL

c)

Swallows	12	2.667

d)

Dragons	2	3.5

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

The groups are based on values in the Opponent column. The Swallows appear there twice, so COUNT(*) for this group is 2. The average number of runs in those two rows is 4.5. The Dragons also appear twice in the Opponent column, so COUNT(*) for this group is also 2. Only one of those rows has a non-NULL Runs value, so AVG(Runs) for this group is just the non-NULL value, 7.

The other four teams appear as an Opponent in only one row, so COUNT(*) is 1 for those groups. For the Bay Stars, Tigers, and Giants, there is a non-NULL value for Runs, so AVG(Runs) is that value. However, the one row for the Carp has NULL in the Runs column. An average (or any aggregation) over an empty list of values is NULL, not 0.

Notice that four rows have NULL in the Opponent column. These form a seventh group, with COUNT(*) = 4. Since there are no non-NULL values of Runs for this group, the value of AVG(Runs) is NULL. Note that none of the choices were about this group.

Opponent	COUNT(*)	AVG(Runs)
Swallows	2	4.5
Bay Stars	1	9
Dragons	2	7
Tigers	1	2
Carp	1	NULL
Giants	1	0
NULL	4	NULL

6. Here are three relations, R(A,B), S(C,D), and T(E,F). Their current values are:

R

A	B
0	1
1	0
1	1

S

C	D
0	1
1	0
1	1

T

E	F
0	1
1	0
1	1

Compute the result of the query:

```
SELECT A, F, SUM(C), SUM(D)
FROM R, S, T
```

```
    WHERE B = C AND D = E
    GROUP BY A, F
    HAVING COUNT(*) > 1
```

Identify, in the list below, the row that appears in the result.

a) (0,0,1,1)

b) (1,1,2,1)

c) (0,1,2,2)

d) (1,1,2,2)

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

Row (1,1) of S joins with two rows --- (0,1) and (1,1) --- of R and with two rows --- (1,0) and (1,1) --- of T, yielding four rows in the result. Row (0,1) of S joins with only (1,0) of R, but two rows of T: (1,0) and (1,1). Likewise, (1,0) if S joins with two rows of R --- (0,1) and (1,1) --- but only one row of T: (0,1). There are thus eight rows in the result of the select-from-where part. We show these eight rows below, grouped by their A and F values.

A	B	C	D	E	F
0	1	1	1	1	0

0	1	1	1	1	1
0	1	1	0	0	1

1	1	1	1	1	0
1	0	0	1	1	0

1	1	1	1	1	1
1	0	0	1	1	1
1	1	1	0	0	1

Finally, we must apply the HAVING clause. That clause says the group must have more than one row. Therefore, the first group, corresponding to A=F=0, is eliminated. However, the remaining three groups survive. The three resulting rows are:

A	F	SUM(C)	SUM(D)
0	1	2	1
1	0	1	2
1	1	2	2

7. The table Arc(x,y) currently has the following tuples (note there are duplicates): (1,2), (1,2), (2,3), (3,4), (3,4), (4,1), (4,1), (4,1), (4,2). Compute the result of the query:

```
SELECT a1.x, a2.y, COUNT(*)
FROM Arc a1, Arc a2
WHERE a1.y = a2.x
GROUP BY a1.x, a2.y;
```

Which of the following tuples is in the result?

a) (1,3,2)

b) (4,3,3)

c) (3,4,2)

d) (3,2,6)

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

The query is asking for the number of paths of length 2 between each pair of nodes a1.x and a2.y. To compute these numbers, we can theta-join Arc with itself, with the condition that the second component of the first copy equal the first component of the second copy. Here is a list of all the tuples and their multiplicities that we get:

- (1,2,2,3) [twice]
- (2,3,3,4) [4 times]
- (3,4,4,1) [6 times]
- (3,4,4,2) [twice]
- (4,1,1,2) [6 times]
- (4,2,2,3) [once]

Since no two of these tuples have the same first and 4th components, we do not have to combine any of these six groups to get the grouping according to the SQL GROUP-BY. Thus, the tuples of the result are formed from the first and fourth components and the multiplicity of the tuple.

8. Suppose relations R(A,B) and S(B,C,D) are as follows:

R =

A	B
1	2
3	4
5	6

S =

B	C	D
2	4	6
4	6	8
4	7	9

Compute the result of the following query:

```
SELECT R.A, R.B, S.B, S.C, S.D
FROM R OUTER JOIN S
    ON (R.A > S.B AND R.B = S.C)
```

Which of the following tuples of R or S is dangling (and therefore needs to be padded in the outer join)?

a) (4,7,9) of S.

b) (4,6,8) of S.

c) (3,4) of R.

d) (2,4,6) of S.

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Let us begin by computing the theta-join. Tuple (3,4) from R(A,B) joins with (2,4,6) from S(B,C,D). The reason is that R.A (=3) > S.B (=2) and R.B = S.C

(=4). Also, (5,6) from R joins with (4,6,8) from S. The reason is that R.A (=5) > S.B (=4) and R.B = S.C (=6). However, no other tuples from R and S join.

Now, to find the dangling tuples, observe which tuples were not mentioned in a successful join; these are (1,2) from R and (4,7,9) from S. We need to pad (1,2) to make (1,2,NULL,NULL,NULL) and we pad (4,7,9) to make (NULL,NULL,4,7,9).

---

9. The table

Scores(Team, Day, Opponent, Runs)

Gives the scores in the Japanese Baseball League for two consecutive days. The Opponent is NULL if the Team did not play on that day. The number of Runs is given as NULL if either the team did not play, or will play on that day but the game is not yet concluded. The data in this table is as follows:

Team	Day	Opponent	Runs
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	NULL	NULL
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	NULL	NULL
Dragons	Monday	Carp	NULL
Tigers	Monday	NULL	NULL
Carp	Monday	Dragons	NULL
Swallows	Monday	Giants	0
Bay Stars	Monday	NULL	NULL
Giants	Monday	Swallows	5

Consider the following three queries, which differ only in the team mentioned in the WHERE clause.

```
1)   SELECT COUNT(*), COUNT(Runs)
     FROM Scores
     WHERE Team = 'Tigers'

2)   SELECT COUNT(*), COUNT(Runs)
     FROM Scores
     WHERE Team = 'Carp'

3)   SELECT COUNT(*), COUNT(Runs)
     FROM Scores
     WHERE Team = 'Swallows'
```

Then, identify in the list below the result of one of these queries.

a) For the Swallows:

2	2

b) For the Carp:

1	0

c) For the Tigers:

12	2

d) For the Tigers:

1	1

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Each of the queries selects two rows of the table; for example, query (1) selects rows 2 and 8. Thus, COUNT(*) produces 2 for each query. COUNT(Runs) produces the number of rows that do not have NULL in the Runs column.

For query (1), only row 2 does not have NULL, so the tuple produced is:

2	1

For query (2), both matching tuples, 3 and 9, have NULL in the Runs column. When there is nothing over which to take an aggregate, the result is NULL, with the exception of the COUNT operator, where the count is then 0. Thus, query (2) produces:

2	0

For query (3), both matching rows, 4 and 10, are non-NULL in the Runs column, so COUNT(Runs) is 2. The tuple produced is:

2	2