Naveen

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

Help

Homework Assignment Submitted Successfully.

You obtained a score of 0.0 points, out of a possible 24.0 points.
You have answered 2 questions correctly.
You have answered 6 questions incorrectly.
For each correct answer, you received 3.0 points
and for each incorrect answer, you lost 1.0 points.

Please Try Again.

Submission number:	375374
Submission certificate:	JE677571
Submission time:	2017-11-03 17:46:09 PST (GMT - 8:00)

Number of questions:	8
Positive points per question:	3.0
Negative points per question:	1.0
Your score:	0

Aggregates, GROUP BY, HAVING, transactions, etc.

1. The relation R(x) consists of a set of integers --- that is, one-component tuples with an integer component. Alice's transaction is a query:

```
SELECT SUM(x) FROM R;
COMMIT;
```

Betty's transaction is a sequence of inserts:

```
INSERT INTO R VALUES(10);
INSERT INTO R VALUES(20);
INSERT INTO R VALUES(30);
COMMIT;
```

Carol's transaction is a sequence of deletes:

```
DELETE FROM R WHERE x=30;
DELETE FROM R WHERE x=20;
COMMIT;
```

Before any of these transactions execute, the sum of the integers in R is 1000, and none of these integers are 10, 20, or 30. If Alice's, Betty's, and Carol's transactions run at about the same time, and each runs under isolation level `READ COMMITTED`, which sums could be produced by Alice's transaction? Identify one of those sums from the list below.

a) 1080

b) 990

c) 1060

d) 950

Answer submitted: **d)**

Your answer is incorrect.

Since 20 and 30 are not in R before the transactions run, deletion of these values can only affect the sum seen by Alice if the number had been inserted by Betty. Transactions in SQL are discussed in Section 6.6 (p. 296). The READ COMMITTED isolation level is defined in Section 6.6.6 (p. 304).

2. The latest scores from the Japanese Baseball League are in the table with schema

Scores(Team, Opponent, RunsFor, RunsAgainst)

The data in this table is as follows:

Team	Opponent	RunsFor	RunsAgainst
Dragons	Tigers	5	3
Carp	Swallows	4	6
Bay Stars	Giants	2	1
Marines	Hawks	5	3
Ham Fighters	Buffaloes	1	6
Lions	Golden Eagles	8	12
Tigers	Dragons	3	5
Swallows	Carp	6	4
Giants	Bay Stars	1	2
Hawks	Marines	3	5
Buffaloes	Ham Fighters	6	1
Golden Eagles	Lions	12	8

What is the result of executing on this data the query:

```
SELECT Team
FROM Scores
WHERE RunsFor > RunsAgainst AND
    RunsFor <= RunsAgainst + 2
```

Identify in the list below, the team that appears in the output.

a) Marines

b) Hawks

c) Golden Eagles

d) Carp

Answer submitted: **a)**

You have answered the question correctly.

3. 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)

Dragons	2	3.5

b)

Carp	1	NULL

c)

Tigers	2	2

d)

Bay Stars	12	2.667

Answer submitted: **a)**

Your answer is incorrect.

Although there are two rows with "Dragons" in the Opponent column, one of them has NULL in the Runs column. A NULL does not count as 0 in an average; it is not counted at all. Grouping is the subject of Section 6.4.5 (p. 285). See also Section 6.4.6 (p. 287), which explains how NULL is handled.

4. Relation R has schema:

```
CREATE TABLE R (
    a INT PRIMARY KEY,
    b INT DEFAULT 0,
    c INT NOT NULL
);
```

R is currently empty. Develop a test that determines whether an insertion into R is currently legal. Then apply your test to determine which of the following INSERT statements is allowable.

a) INSERT INTO R(c) VALUES(0);

b) INSERT INTO R(a,c) VALUES(1,1);

c) INSERT INTO R VALUES(1,NULL,2);

d) INSERT INTO R(b,c) VALUES(3,4);

Answer submitted: **d)**

Your answer is incorrect.

It fails to give a non-NULL value to the primary key. This question involves a number of different concepts. You can find what you need among the following sections: NULL values (6.1.6, p. 252), INSERT statements (6.5.1, p. 291), default values (2.3.5, p. 34), primary keys (2.3.6, p. 34), and not-NULL constraints (7.2.1, p. 319).

5. Consider the following transactions:

- S: [X := X + 10; Y := Y - 10]
- T: [X := X * 2; Y := Y * 2]
- U: [Y := Y + 10; X := X - 10]

Assuming initial values of X = 15 and Y = 25, serializable schedules of these three transactions can leave the database in various states. Determine all such possible states. Then, identify from the list below, which of the following is a possible state of the database resulting from a serializable execution of S, T and U.

a) X = 30; Y = 60

b) X = 25; Y = 15

c) X = 20; Y = 70

d) X = 20; Y = 60

Answer submitted: **a)**

Your answer is incorrect.

Hint: remember that the effect of any serializable schedule must be the same as that of some serial schedule. You may wish to consult Sect. 17.1.3 (p. 846) on database states, and Sect. 18.1.3 (p. 886) on serializable schedules.

6. Suppose relation R(a,b,c) has the following tuples: (1,1,3), (1,2,3), (2,1,4), (2,3,5), (2,4,1), (3,2,4), and (3,3,6). Define the view *V* by:

```
CREATE VIEW V AS
    SELECT a+b AS d, c FROM R;
```

What is the result of the query:

```
SELECT d, SUM(c) FROM V
GROUP BY d
HAVING COUNT(*) <> 1;
```

Identify, from the list below, a tuple in the result.

a) (1,8)

b) (2,7)

c) (6,7)

d) (6,9)

Answer submitted: **b)**

Your answer is incorrect.

Hint: The view V(d,c) is a set of pairs formed from the triples in relation R. Each tuple of V has the first component equal to the sum of the first two components of a tuple of R, while the second component of the tuple of V is the third component of the same tuple of R. Compute V first, and then apply the query to that relation. Relevant reading: grouping and aggregation, Sections 6.4.4 and 6.4.5 (p. 284), and views, Section 8.1 (p. 341).

7. 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) COUNT(A)

b) D

c) B*E

d) B+E

Answer submitted: **b)**

You have answered the question correctly.

8. Consider the following database schema and the five queries labelled Q_1,...,Q_5. Assume that all movies released in any given year have distinct titles .

movie(title, year, length, in_color, studio_name, producer_name)
stars_in(movie_title, movie_year,star_name)
movie_star(name, address, gender, birthdate)
movie_producer(name, address, net_worth)
studio(name, address)

Q_1: Who were the male stars in *Terms of Endearment* released in 1983?

Q_2: Which movies are longer than *Gone With the Wind*

Q_3: Find the titles which have been used for two or more movies?

Q_4: Find the producers of *Harrison Ford*'s movies.

Q_5: Find the total length of all movies for only those producers who made at least one film prior to 1930.

Identify the correct SQL statement corresponding to these queries.

a)

Q_3:

```
SELECT DISTINCT movie.title
FROM movie,movie AS mov
WHERE movie.title = mov.title
```

b)

Q_5:

```
SELECT producer_name, SUM(length)
FROM movie
WHERE MIN(year) > 1930
GROUP BY producer_name
```

c)

Q_4

```
SELECT producer_name
FROM stars_in,movie
WHERE star_name = 'Harrison Ford' AND movie_title = title
```

d)

Q_3:
```
SELECT DISTINCT movie.title
FROM movie,movie AS mov
WHERE movie.title = mov.title AND movie.year ≠ mov.year
```

Answer submitted: **c)**

Your answer is incorrect.

You need to identify the primary key of the relation movie correctly.