

Quiz 3

The following questions deal with SQL, PLpgSQL, aggregates.

Deadline	Saturday, 26 October 2019 at 10:00PM
Latest Submission	Thursday, 24 October 2019 at 2:10PM
Raw Mark	4.00/4.00 (100.00%)
Late Penalty	N/A
Final Mark	4.00/4.00 (100.00%)

Question 1 (1 mark)

Consider a table built with the following SQL statements

```
create table t (x integer, y integer);
insert into T values
(1,2), (2,3), (3,4), (1,5), (2,6), (3,7), (1,8), (2,9), (3,10);
```

And consider the following query on this table:

```
select x, sum(y) from T group by x;
```

What are the values in the **sum** column of the results table:

x	sum
1	<input type="text" value="15"/>
2	<input type="text" value="18"/>
3	<input type="text" value="21"/>

✓ Your response was correct.

Mark: 0.33 + 0.33 + 0.33 = 1.00

Question 2 (1 mark)

Consider a database with one table `enrolments` defined as:

```
create table enrolments (
    student text,
    course text,
    mark integer check (mark between 0 and 100),
    grade char(1) check (grade between 'A' and 'E'),
    primary key (student,course)
);
```

and containing the following data:

```
db=# select * from enrolments order by course;
```

student	course	mark	grade
james	COMP1917 12s1	50	D
peter	COMP1917 12s1	45	E
john	COMP1917 12s1	90	A
peter	COMP1917 12s2	40	E
john	COMP1927 12s2	85	A
james	COMP1927 12s2	55	D
james	COMP2911 13s1	50	D
john	COMP2911 13s1	85	A
john	COMP3311 13s2	70	B

Now consider the following type and PLpgSQL function:

```
create type stu_res as
    (student text, score numeric(5,2));
create function results() returns setof stu_res
as $$
declare
    r record; res stu_res;
    p text := ''; s integer := 0; n integer := 0;
begin
    for r in
        select student,mark
        from enrolments
        order by student
    loop
        if (p <> r.student and n > 0) then
            res.student := p;
            res.score := (s::float/n)::numeric(5,2);
            return next res;
            s := 0; n := 0;
        end if;
        n := n + 1;
        s := s + r.mark;
        p := r.student;
    end loop;
    if (n > 0) then
        res.student := r.student;
        res.score := (s::float/n)::numeric(5,2);
        return next res;
    end if;
end;
$$ language plpgsql;
```

Which of the following gives the output of the SQL statement:

```
db=# select * from results();
```

(a) ☐

```
student | score
-----+-----
(0 rows)
```

(b) ☒

```
student | score
-----+-----
james   | 51.67
john    | 82.50
peter   | 42.50
(3 rows)
```

(c) ☐

```
student | score
-----+-----
james   | 155.00
john    | 330.00
peter   | 85.00
(3 rows)
```

(d) ☐

```
student | score
-----+-----
james   | 55
james   | 50
james   | 50
john    | 70
john    | 90
john    | 85
john    | 85
peter   | 45
peter   | 40
(9 rows)
```

(e) ☐

None of the other answers is correct.

✓ Your response was correct.

Mark: 1.00

Question 3 (1 mark)

Which of the following SQL statements could be used to produce the same result as the output from the previous question?

(a) <input type="radio"/>	<pre>select student, mark as score from enrolments order by student;</pre>
(b) <input type="radio"/>	<pre>select student, sum(mark)::numeric(5,2) as score from enrolments group by student order by student;</pre>
(c) <input checked="" type="radio"/>	<pre>select student, avg(mark)::numeric(5,2) as score from enrolments group by student order by student;</pre>
(d) <input type="radio"/>	<pre>select student, mark::numeric(5,2) as score from enrolments group by student order by student;</pre>
(e) <input type="radio"/>	None of the other answers is correct.

✓ Your response was correct.

Mark: 1.00

Question 4 (1 mark)

Consider the following aggregate definition:

```

create type IntPair as (x integer, y integer);

create function
    next_state(p IntPair, n integer) returns IntPair
as $$
begin
    if (p.x is null) then
        p.x := n;
    elsif (p.y is null) then
        if (n < p.x) then
            p.y := n;
        elsif (n > p.x) then
            p.y := p.x; p.x := n;
        end if;
    elsif (n > p.x) then
        p.y := p.x; p.x := n;
    elsif (n < p.x and n > p.y) then
        p.y := n;
    end if;
    return p;
end;
$$ language plpgsql;

create function
    second(p IntPair) returns integer
as $$
begin
    return p.y;
end;
$$ language plpgsql;

create aggregate max2 (int) (
    sfunc = next_state,
    stype = IntPair,
    finalfunc = second
);

```

What is the output of the following SQL statement applying this aggregate to the enrolments table from question 2?

```
db=# select max2(mark) from enrolments;
```

(a) <input type="radio"/>	63.33
(b) <input type="radio"/>	70
(c) <input checked="" type="radio"/>	85
(d) <input type="radio"/>	90
(e) <input type="radio"/>	(90,85)

(f) <input type="radio"/>	(90,40)
(g) <input type="radio"/>	None of the other answers is correct.

✓ Your response was correct.

Mark: 1.00