Quiz 4

Deadline	Saturday, 09 November 2019 at 12:06AM
Latest Submission	Friday, 08 November 2019 at 4:15PM
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 the following tables:

```
create table S (x integer, y integer);
create table T (z integer);
```

and an assertion that holds between them

```
all S.x values must be larger than all T.z values
```

A trigger function to check this might look like:

```
create or replace function checkXlarger() returns trigger
as $$
declare
    minX integer; maxZ integer;
begin
    select min(x) into minX from S;
    select max(z) into maxZ from T;
    if (maxZ >= minX) then
       raise exception 'All S.x must be greater than all T.z';
    end if;
    return new;
end;
$$ language plpgsql;
```

If we insert this function into the database, what would be the *most appropriate* way to set up the trigger? Assume that we also want all of the standard constraint checks to be carried out.

(a)

```
create trigger checkAssert before insert or update on S
for each row execute procedure checkXlarger();
```

(b)	`create trigger checkAssert after insert or update on S` `for each row execute procedure checkXlarger();`

(c)	`create trigger checkAssert1 before insert or update on S`
	`for each row execute procedure checkXlarger();`
	`create trigger checkAssert2 before insert or update on T`
	`for each row execute procedure checkXlarger();`

(d)	`create trigger checkAssert1 after insert or update on S`
	`for each row execute procedure checkXlarger();`
	`create trigger checkAssert2 after insert or update on T`
	`for each row execute procedure checkXlarger();`

(e)	No trigger is needed; this whole question is a bluff. This checking could be done more simply with standard database constraints

✔ Your response was correct.
Mark: 1.00

Question 2 (1 mark)

I could define an aggregate to concatenate a column of strings (like `string_agg()` but without the delimiter), as follows

```
create aggregate concat(text) (
    sfunc = append,
    stype = text,
    initcond =
 ''
 ,
);
```

where the `append()` function simply appends one string to another (e.g. using `x||y`).

Fill in the blank to indicate a suitable value for `initcond`?

✔ Your response was correct.
Mark: 1.00

Question 3 (1 mark)

To connect to a database called `mydb` on Grieg, I would use the following Python statement

```
conn = psycopg2.connect("
 dbname=mydb
 ")
```

Fill in the minimal database connection string that would accomplish this.

✔ Your response was correct.
Mark: 1.00

Question 4 (1 mark)

Assuming that we have successfully established a database connection in an object called `conn`, and that we have a database containing the following table called **R**

```
 x | y | z
---+---+---------
 1 | 2 | first
 2 | 3 | second
 3 | 4 | third
 1 | 5 | fourth
 2 | 6 | fifth
 3 | 7 | sixth
 1 | 1 | odd-one
```

What is printed by the following python code:

```
db = conn.cursor()
db.execute("select * from R")
s = 0
for t in db.fetchall():
  a, b, c = t
  s = s + a*b
print(s)
```

```
59
```

✔ Your response was correct.
Mark: 1.00