WEEK 5 CHALLENGE SOLUTIONS

Below are suggested answers. Note, in some cases, there's more than one way of getting the correct answer so check to see if your codes will produce the same results as the codes below

1. What is the average amount spent by a subset of customers *who paid more than* the **average amount spent by all the customers in the database**? Make sure to round your answer to two decimal places.

Note: The average amount should be computed for all payment transactions regardless of the number of transactions or amount paid per customer.

Solutions:

Step 1: Compute the average amount paid by all customers.

```
SELECT AVG(amount)
FROM payment
```

Step 2: Select or show all the amount spent by customers who paid more than the **average amount spent by all customers** (computed in step 1)

```
SELECT amount
FROM payment
WHERE amount > (SELECT AVG(amount)
                FROM payment);
```

Step 3: Compute the average amount spent by a subset of customers obtained from step 2.

```
SELECT ROUND(AVG(amount),2) AS AVG_subset
FROM ( SELECT amount
       FROM payment
       WHERE amount > (SELECT AVG(amount)
                       FROM payment) ) sub1;
```

2. Show the film's title, its description and another attribute that you need to create called **string_film**. Under this attribute, if the title starts with 'A', extract or show the first word in the title. Otherwise, write 'NA' (which stands for 'not applicable').

```
SELECT title, description, CASE WHEN LEFT(title, 1) = 'A' THEN
LEFT(title, POSITION(' ' IN title)-1)
    ELSE 'NA'
    END AS string_film
FROM film;
```

3. In the email attribute, extract the 'sakilacustomer.org' and call this new attribute web_address.

```
SELECT SUBSTRING(email, POSITION('@' IN email)+1, LENGTH(email))
AS web_address
FROM customer;
```

4. The company has a holiday special. The special promo is to charge only the integer part of the amount (or to remove the decimal part). For example, if the amount is $7.99, the customer will be charged $7 only.

Come up with this new column that shows this new price (call it holiday_promo).

Hint: You may refer to the amount attribute in payment table. Now since the amount is **numeric**, you need to **first convert it to string** using a function called CAST and specify the number of characters --- for #.## format it would be 4 characters). It looks like the one below and you can put it inside another function.

CAST(amount AS character varying(4))

```
SELECT LEFT(CAST(amount AS character varying(4)), 1)
FROM payment;
```