



## Assignment 6 (08.06.2022)

Handin until: 24.06.2022, 09:00

Until June 15th 2022, students have the opportunity to **evaluate lectures**. Please help us to improve **your** courses by providing precious feedback. Check your Mailbox **now** to participate.

### 1. [6 Points] Sorted Arrays

```
1 CREATE TABLE arrays (
2   id  int GENERATED ALWAYS AS IDENTITY,
3   arr int[] CHECK (cardinality(arr) > 0)
4 );
```

```
1 INSERT INTO arrays(arr) VALUES
2   (ARRAY[1,2,3,5,6]),
3   (ARRAY[4,3]),
4   (ARRAY[7,6,9]),
5   (ARRAY[2,2]);
```

Write a SQL query that scans table `arrays` for arrays which are **sorted in ascending order** and lists them as a table `result(id, arr)`.

**Example:** The result for the sample instance above is

id	arr
1	{1,2,3,5,6}
4	{2,2}

### 2. [7 Points] Lights

A hallway is lit by  $N \geq 1$  light bulbs on the ceiling. Each bulb has a pull cord which turns its light on and off. Initially, all lights are off.  $N$  people go down the hallway (in the original problem  $N$  is 43):

- The first person pulls all cords 1,2,3, ...
- The second person pulls every other cord: 2,4,6, ...
- The third person pulls every third cord: 3,6,9, ...
- This continues until, finally, the 43rd person pulls cord 43 only.

Which bulbs are on after all people have passed through the hallway?

Formulate a SQL query to compute a single-column table which list the numbers of all lit bulbs after all people passed through the hallway.

**Note:** Your SQL code has to work for any  $N \geq 1$ . Please make  $N$  configurable using `\set N ....`

### 3. [7 Points] Data Analysis

We provide you with a CSV file `data.csv` which encodes five data sets (see the first column) of points  $(x,y)$  in the 2D plane. First, load the data from the CSV file into a table `analysis`.

```
1 -- Define result as type alias for numeric(3,1).
2 CREATE DOMAIN result AS numeric(3,1);
3
4 CREATE TABLE analysis (
5     dataset char(1) NOT NULL,
6     x       numeric NOT NULL,
7     y       numeric NOT NULL
8 );
9
10 \copy analysis FROM '/path/to/data.csv' WITH (FORMAT csv, HEADER TRUE);
```

- (a) Write a SQL query which calculates the mean of `x` and `y`, the standard deviation of `x` and `y` and the correlation coefficient of columns `x` and `y`<sup>1</sup>. Each of the result columns has to be of domain type `result`. We expect the query to produce a table with following schema:

dataset	x_mean	y_mean	x_stddev	y_stddev	correlation
a	...	...	...	...	...
b	...	...	...	...	...
⋮	...	...	...	...	...

**Note:** If you compare these statistical measures across the five data sets, you may notice something peculiar.

- (b) The statistical measures found in (a) do not tell the whole story of the data sets. Let's make this apparent:

For each data set `a - e` in `data.csv`, generate a 2D plot<sup>2</sup> of its points (label the `x` and `y` axes of the plots). From these plots, it will be easy to derive a descriptive name for each of the five data sets. Download the plots in PDF format and use the descriptive names when you name and hand in the five `.pdf` files.

### 4. [10 Points] SEND + MORE = MONEY

Solve this digit assignment (verbal arithmetic, cryptarithm) puzzle, in which each letter is to be replaced by a decimal digit `0,1,...,9` such that the arithmetic works out:

```
  SEND
+ MORE
-----
= MONEY
```

- (a) First, write a SQL user-defined function `val(digits int[])` which converts the array of integers `digits` into its actual integer representation and returns it.

Examples:

- `val(array[1,2,3,4]) ≡ 1234`
- `val(array[]) ≡ NULL`

- (b) With the help of `val(digits int[])`, write a SQL query which solves the digit assignment puzzle. Return a table with eight columns `S, E, N, D, M, O, R, Y` of type `int` such that each column holds the letter's digit replacement. If the puzzle has  $n$  distinct solutions, your table must also have  $n$  rows.

Hints:

- Assigned digits must be unique.
- `M` obviously must be 1.
- Brute force is a legitimate approach but may take a few minutes to compute.

<sup>1</sup><https://www.postgresql.org/docs/current/functions-aggregate.html#FUNCTIONS-AGGREGATE-STATISTICS-TABLE>

<sup>2</sup><https://huygens.science.uva.nl/PlotTwist/>