Advanced SQL
Forum: `https://forum-db.informatik.uni-tuebingen.de/c/ss20-asql`

Assignment 6

Relevant videos: up to #27
`https://tinyurl.com/AdvSQL-2020`
Submission: Tuesday, 16.06.2020, 10:00 AM

1. [6 Points] **Sorted Arrays**

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

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

Write a SQL query to scan table `arrays` for arrays which are **sorted in ascending order** and return them as a table `sorted(id, arr)`.

Example: The result of your query in the example above is:

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

2. [7 Points] **Mountain Altitudes**

```
CREATE TABLE mountains (
  y int  GENERATED ALWAYS AS IDENTITY,
  m text
);
```

```
INSERT INTO mountains(m) VALUES
  (' # # '),
  ('### '),
  ('####');
```

Write a SQL query to "parse" table `mountains` and return a table `map(x,alt)` (as seen in Chapter 5, Slide 14). One `#` measures 100 meters in height. You can assume that each vertical mountain segment is one continuous pile of `#` starting at the very bottom *i.e.*, the minimum `alt` value is `100`. The value of `x` ascends from left to right, starting at `1`.

Example: The result of your query in the example above is:

x	alt
1	200
2	300
3	200
4	100

3. [7 Points] **Data Analysis**

We provided you with a CSV file `data.csv` which encodes five data sets (see the first column) of points (x,y) in the 2D plane. To be able to analyse each data set, you have to load the data inside the CSV file into a table `Analysis`.

```
-- Define result as type alias for numeric(3,1).
CREATE DOMAIN result AS numeric(3,1);

CREATE TABLE analysis (
  dataset char(1) NOT NULL,
  x       numeric NOT NULL,
  y       numeric NOT NULL
);

\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`[1]. Each of the results has to be of type `result`, as previously defined. Your query should produce the following result:

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) Plot[2] each data set of `data.csv` in a 2D plane to derive a descriptive name in place of the *bland* `a`, `b`, `c`, `d` and `e`.

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(ds int[])` which converts the array of integers `ds` 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(ds 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 should 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.

[1] `https://www.postgresql.org/docs/12/functions-aggregate.html#FUNCTIONS-AGGREGATE-STATISTICS-TABLE`
[2] Plotting software: `https://plot.ly/create/`, `http://gnuplot.respawned.com/`.