



Assignment 7 (24.06.2022)

Handin until: 01.07.2022, 09:00

Important: We assume all the queries you write for this assignment most probably will utilize **window functions**.

1. [10 Points] Advent of Code

Advent of Code (AoC) is an Advent Calendar with a collection of small programming puzzles for each day in December until 25th of December. For this task, you will solve the puzzle of AoC day 1 from 2021¹. Read the AoC day 1 puzzle description carefully.

- (a) First, log into AoC and download the puzzle **input** file. Then, load it into table **puzzle_input** with the following query:

```
1 CREATE TABLE puzzle_input (
2   line int PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
3   val int
4 );
5 \copy puzzle_input(val) FROM '/path/to/input';
```

Don't forget to submit your puzzle **input** file.

- (b) Write a SQL query to solve the first part of the AoC day 1 puzzle. The result of the SQL query is a table with exactly one integer: the number of depth increases.
- (c) [5 Bonus Points] Write a SQL query to solve the second part of the AoC task. Submit the number of depth increases of (b) to reveal the second part of the puzzle. The result of the SQL query is a table with exactly one integer: the number of sums that are larger than the previous sum.

2. [10 Points] Seesaw Balance

We want to figure out where to position the pivot to try and balance a seesaw. As such we provided you with a table **seesaw** populated by 100 random weights uniquely identifiable by their position **pos** $\in (1, 2, \dots, 100)$ on the seesaw:

```
1 CREATE TABLE seesaw (
2   pos int GENERATED ALWAYS AS IDENTITY,
3   weight int NOT NULL
4 );
```

```
1 INSERT INTO seesaw(weight) (
2   SELECT floor(random()*10) AS weight
3   FROM generate_series(1,100) AS _
4 );
```

Write a SQL query determining which positions **pos** would be best to balance the seesaw. To balance the seesaw optimally we have to determine the minimal weight difference between the sum of the weights left and right of the pivot. The result of your query produces a table with a column **pos**, which holds the positions at which to place the pivot, and a column **diff**, which holds the difference between the sum of the weights left and right of the pivot placed at **pos**.

¹<https://adventofcode.com/2021/day/1>

Simplified Example:

weight	1	5	3	4	3	7	6	9	3
pos	1	2	3	4	5	6	7	8	9

Let us assume a seesaw with only 9 positions. We find that, at **pos** = 6, the difference between the sum of the weights left and right of the pivot is $|16 - 18| = 2$, which is minimal. The result of the query for this seesaw is therefore:

pos	diff
6	2

3. [10 Points] Replace NULL

The physicist from assignment 2 needs your help again. Since then, she caught up on various aspects of RDBMSs and hands you a measurement table defined as follows:

```
1 CREATE TABLE measurements (  
2   ts timestamp PRIMARY KEY,  
3   val numeric  
4 );
```

She explains: every measure has its own unique **timestamp** and is represented as a **numeric** value. But some measurements were inconclusive and, as such, are represented as **NULL**.

Example: A possible set of measurements may look as follows:

ts	val
2019-12-04 07:34:59	NULL
2019-12-04 07:37:16	42.0
2019-12-04 07:38:36	4.1
2019-12-04 07:42:33	NULL
2019-12-04 07:55:06	NULL
2019-12-04 07:57:06	12.3
2019-12-04 08:03:18	NULL
2019-12-04 08:15:44	15.1
2019-12-04 08:22:21	2.2
2019-12-04 08:37:31	NULL

She then asks you to write a SQL query which replaces each **NULL** value with the previous most recent conclusive measurement. She also tells you to drop any leading inconclusive measurements. For the example above, your query should produce the following result:

ts	val
2019-12-04 07:37:16	42.0
2019-12-04 07:38:36	4.1
2019-12-04 07:42:33	4.1
2019-12-04 07:55:06	4.1
2019-12-04 07:57:06	12.3
2019-12-04 08:03:18	12.3
2019-12-04 08:15:44	15.1
2019-12-04 08:22:21	2.2
2019-12-04 08:37:31	2.2