



Assignment 9

Hand in this assignment until Friday, January 9th 2025, 12:00 pm at the latest.

🤔 Running out of ideas?

Are you hitting a roadblock? Are some of the exercises unclear? Do you just need that one hint to get the ball rolling? Refer to the [#forum](#) channel on our Discord server—maybe you'll find just the help you need.

🎉 Happy Holidays!

The entire Advanced SQL team wishes you a very happy holiday season! We hope you enjoy this time and get some rest. Don't stress about this exercise sheet; you have until after the new year to complete it. 😊

💡 Note

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

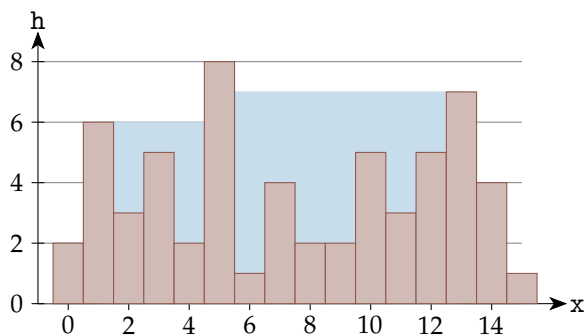
📝 Exam-style Exercises

Exercises marked with 📝 are similar in style to those you will find in the exam. You can use these to hone your expectations and gauge your skills.

Task 1: Water Bars

WINDOW FUNCTIONS

You are given a table of bars specified by their position x and their height h .



```
1 CREATE TABLE bars (
2   x int,
3   h int
4 );
```

Write a SQL query that computes the amount of water the bar chart can hold as depicted above. The result of the SQL query is a table with exactly one integer: for the given example you should receive 35.

💡 Hint

Since, obviously, the heights of neighbouring bars are relevant here, consider to use *scan*-like window functions to tackle this task.

Task 2: Bike Tour

WINDOW FUNCTIONS

We recorded a dataset of waypoints and elevation of a biking tour from Rottenburg to the WSI. The dataset is provided in table `tour` of file `tour.sql`. This file also contains a macro `distance` which calculates the earth distance between two waypoints in meter. A waypoint is defined through its latitude and longitude.

Write SQL queries to calculate the following:

- A** For each waypoint w , calculate the distance from home (Rottenburg, the first data point) to w as the crow flies, i.e., the direct earth distance between home and the waypoint. The result lists the `id` of w and the **earth distance** between home and the waypoint w .
- B** For each waypoint w , calculate the cycled (waypoint to waypoint) distance from home to w . The result lists the `id` of w and the **cycled distance** between home and the waypoint.
- C** For each waypoint w , calculate the *estimated* grade of the slope at w from the height differences of the waypoints just before/after w . The result lists the `id` of w and **slope** in %.

Task 3: Consecutive Logins

WINDOW FUNCTIONS

We define a table `work` as follows:

```
1 CREATE TABLE work (  
2   emp_id integer,  
3   login date  
4 );
```

<code>work</code>	
<code>emp_id</code>	<code>login</code>
2	'2016-04-06'
4	'2016-04-06'
2	'2016-04-06'
4	'2016-04-07'
2	'2016-04-07'
5	'2016-04-07'
2	'2016-04-10'
5	'2016-04-08'
2	'2016-04-11'
5	'2016-04-09'

Each row (e, d) in table `work` indicates that employee e was logged in on day d .

Write a SQL query that returns a table with schema $(\text{emp_id}, \text{streak})$, where for each employee (`emp_id`), `streak` is the maximum number of consecutive days that employee was logged in.

Example

For the sample logins given in table `work`, the query results in a table where the employees with `id = 2` and `id = 4` had, at most, 2 consecutive days of logins and the employee with `id = 5` had, at most, 3 consecutive days of logins.

<code>emp_id</code>	<code>streak</code>
2	2
4	2
5	3