

CS-A1153 Databases, Homework 5

Deadline: 25 May 2022, at 18:00 (late submission until 1 June 2022, at 18:00 with 50% of the points)

Please, submit your solutions to all these problems as one zip file containing the .py files and either text or pdf file containing an example execution of the Python program. [Submission box can be found in A+.](#)

[Data set to use in this exercise can be found here.](#)

Original data is downloaded from [Finnish Meteorological Institute](#).

SQL data analysis with Python

1. (5 p.) Write a Python program which executes the following operations using sqlalchemy and pandas libraries. In this exercise you should **not** use the function `run sql from file`, but rather write the SQL commands inside the code.

- Create a new SQLite database with name weatherdata.db and connects to it.
- Create an engine object to use the database.
- Create tables `Place`, `Observation`, and `Temperature` according to the following definitions

```
Place (code, name, latitude, longitude)
Observation (place, date, rain, snow, air temperature, ground temperature)
Temperature (place, date, lowest, highest)
```

The dates should be saved without time (e.g. only "YYYY-MM-DD"). Determine other proper attribute types from the CSV file attached in Problem 2.

The program must output the result of the query to the user. The program does not have to ask any input from the user. Your program does not have to contain any error handling. Include the .py-file containing your Python program with your submission.

2. (15 p.) Consider the [following data set from weather stations](#):

- Year, month, day, and time
- Rain (millimeter, value -1 means no rain, empty means no observation)
- Snow depth (centimeter, value -1 means no snow, empty means no observation)
- The average air temperature (Celcius degree, based usually on 4 or 8 observations per day)
- The average ground temperature (Celcius degree, based usually on 4 or 8 observations per day)
- Highest temperature during previous evening 8 pm and next evening 8 pm (Celcius degree)
- Lowest temperature during previous evening 8 pm and next evening 8 pm (Celcius degree)
- Observation place
- Place code
- Coordinates (latitude and longitude)

Please note, that the CSV includes **raw data** in a single data dump and it needs to be sanitized and separated to different tables. Some data is repeated on every row and observations have been saved in a bit different manner in different places. Also, data is not complete and some observations might be missing.

Our goal is to standardize the observations to our tables (one row per place and date in tables for observations and min/max temperatures). If there are many observations for certain place and date, select suitable method for handling that. (Remember that we are focusing on analyzing trends rather than investing single values, so if you are in a hurry, maybe just selecting one value is enough.)

Continue your work with the Python program you wrote in Problem 1. Now, it is time to make your program to use the input read from a CSV file in SQL statements. Add to your program the following operations using sqlalchemy and pandas libraries:

- Reads the contents of the attached CSV file and inputs it to SQLite database created in the problem 1.
- Sanitize the data as following:
 - Observation: rain and snow should have value 0, if there is no rain or snow. If the rain/snow observation is missing, value should be NULL.
 - If any observation value is missing, the value should be NULL.
 - Temperature: even the minimum/maximum value is measured between previous day 8 pm and current day 8 pm, it can be treated as minimum/maximum for the day the day column determines.

The program must output the result of the query to the user. The program does not have to ask any input from the user. Your program does not have to contain any error handling. Include the .py-file containing your Python program with your submission.

3. (10 p.) Find the answers to the following questions. Include the code you used to find the answers in your submission. You can use SQL queries and the functions for dataframes in pandas library.
 - (a) Find the number of snowy days on each location. Which location (name) has had most snowy days? For this location, find the month with most snow (sum). For the location with least snowy days, find the month with most snowy days.
 - (b) Inspect the rows in Temperature where both "highest" and "lowest" are not NULL. Calculate the [sample correlation coefficient](#) between these two attributes. What can you interpret from this value? Find the correlations when grouping by location.
 - (c) Find out the correlation between average temperature and latitude of the location.
 - (d) For each location, use matplotlib to plot the number of rainy days for each month as a bar plot.
 - (e) For each location, plot the average temperature throughout the year. You may plot all the graphs into the same Figure.