DATABASE MANAGEMENT SYSTEMS, IIITS

Assignment

Instructions:

1. Submit a hard copy (written) assignment with clearly the following written on top of the sheet: Name, Roll No, Section.
2. The assignments are required to be submitted during class hours to the instructor (26-11-2019). No further extensions would be granted.
3. Hard Deadline: 26-11-2019, Tuesday

Q.1: SQL Query Set **[10 points]**

In this problem set, you will write a series of queries using the **SELECT** statement over a table called **expt_table** of light and temperature readings collected from three wireless nodes with special sensing hardware. The data represents a day's worth of light and temperature readings, collected every 30 seconds from each sensor node. Because these sensor that were used are lossy, there are some times when sensor readings are missing. The schema of the sensor data is as follows:

result_time : timestamp	epoch : int	nodeid : int	light : int	temp : int	voltage : int

Where the fields are as follows:

result_time	The time the record was inserted into the database.
epoch	The sensor-node sequence number of the record. Different results from different sensors with the same epoch number should arrive in the database at about the same time. If they have different times, it can indicate a failure in the software that runs on the sensor – one of the queries below asks you to look for such failures.
nodeid	The id of the sensor node that produced this reading. There are three sensors in this data set, numbered 1, 2, and 3.
light	The light reading from the sensor, in raw units relative to the minimum and maximum brightness the sensor can perceive. This is a 10-bit sensor, with a value of 0 corresponding to darkness 1024 representing maximum brightness. The calib_light table maps raw units into Lux, which is a commonly used measure of brightness.
temp	The temperature reading from the sensor, in raw units. The calib_temp table maps raw units into degrees Celsius.
voltage	The voltage on this device, in raw units.

The two tables, **calib_light** and **calib_temp** each have two fields :
raw : integer, **calib** : integer
Where **raw** is the raw sensor value and **calib** is the calibrated value, in Lux (in the case of **calib_light**) or degrees Celsius (in the case of **calib_temp**.)

Questions:
1. Write a query (using the SELECT statement) that will compute times and ids when any sensor's light reading was above 550. Show both the query and the first few lines of the result. **[1 point]**

2. Write a query that will compute the average light reading at sensor 1 between 6 PM and 9 PM (inclusive of 6:00:00 PM and 9:00:00 PM). Show the query and the result. **[1 point]**

3. Write a single query that computes the average temperature and light reading at every sensor between 6 PM and 9 PM, but exclude any sensors whose maximum voltage was greater than 418 during that time period. Show both the query and the result. **[1 point]**

4. Write a query that computes the average calibrated temperature readings from sensor 2 during each hour, inclusive, between 6 PM and 9 PM (i.e., your answer should consist of 4 rows of calibrated temperatures.) **[1 point]**

5. Write a query that computes all the epochs during which the results from sensors 1 and 2 arrived more than 1 second apart. Show the query and the result. Note that you can use the difference (minus) operator on timestamps in Postgres, and that the string '1 second' refers to a period of 1 second. **[1 point]**

6. Write a query that determines epochs during which one or two of the sensors did not return results. Show your query and the first few results, sorted in by epoch number. You may wish to use a nested query – that is, a SELECT statement within the FROM clause of another SELECT statement. **[1 point]**

7. Write a query that produces a temperature reading for each of the three sensors during any epoch in which any sensor produced a reading. If a sensor is missing a value during a given epoch, your result should report the value of this sensor as the most recent previously reported value. If there is no such value (e.g., the first value for a particular sensor is missing), you should return the special value 'null'. You may wish to read about the CASE and OUTER JOIN SQL statements. **[2 points]**

8. Write a query that determines epochs during which all three sensors did not return any results. Note that this is a deceptively hard query to write – you may need to make some assumptions about the frequency of missing epochs. **[2 points]**