**Questions:**

1. **Create a database for this assignment. Execution: 1.5 mark**

The execution screenshot cannot be showed because the database has been created by myself already and I closed the workbench.

**A screenshot of a computer

Description automatically generated**

1. **Instruct MySQL Server to use the database you just created. Execution: 1.5 mark**

**A screenshot of a computer

Description automatically generated**

1. **In a separate SQL tab, open the *barrie\_weather.sql* file provided. Under question 3, copy its contents into your assignment1 script, then run the commands to create the table (also called barrie\_weather, and will be used to answer following questions) and then populate it with data using the provided insert statements. Execution: 2 mark**

Please check the schemas on the left.

**A screenshot of a computer

Description automatically generated**

1. **In the report only, explain how the provided CREATE TABLE statement could be improved. Please focus on the data types and parameters for this discussion. Written answer: 2 marks Bonus: Up to additional 3 marks**
2. Choose the Right Data Types:

* Think about the kind of information you'll be putting in each column, like dates, numbers with decimals, or simple numbers.
* For dates and times, it's better to use a special date-and-time format (like DATETIME) instead of just plain text. It's like using a calendar instead of writing the date on a sticky note.
* If you're storing numbers with lots of decimal points (like temperatures), use data types like FLOAT or DECIMAL to keep those decimals accurate.
* For columns with numbers that don't have many options (like years, months, or small numbers), use data types like TINYINT or SMALLINT. It's like choosing a box that fits your stuff perfectly.

1. Add Rules to Keep Data Safe:

* Imagine you have a checklist for your data. Constraints are like rules on that checklist to make sure everything is correct.
* If a column should always have something in it (no missing values allowed), use a NOT NULL constraint. It's like saying, "You can't leave this part empty."
* To make sure each row is unique and easily identified, you can add a primary key. Think of it as giving each row a special number so you can find it quickly, like a student ID for each student in a class.

1. **The provided INSERT statements in barrie\_weather were generated automatically and load data very slowly because of the way that the script’s commands have been structured. What is it about the structure of this script that causes the script to run slowly? In the report only, explain how you could restructure the commands if you were tasked with inserting the same data into barrie\_weather. Written answer: 3 marks**
2. Single Row Inserts:

The current script operates by inserting data one row at a time, with each INSERT INTO statement handling a single row of data. This method can become considerably sluggish, especially with extensive datasets, due to the necessity of executing numerous individual transactions, resulting in increased overhead and reduced efficiency.

1. Enhancing Efficiency through Restructuring:

To boost the efficiency of data insertion, consider implementing batch or bulk insert techniques. These approaches minimize transaction counts and enhance overall performance. Here's a suggested script restructuring for improved efficiency.

1. **Select all data for every hour of data where the temperature (temp) was greater than 26.5. Provide an additional comment in your SQL script that states the number of rows returned. Execution: 1.5 mark Accuracy: 2 marks**

267 rows returned

**A screenshot of a computer

Description automatically generated**

1. **Select the date\_time, temp, pressure\_kPA and wind\_spd\_kmh for every hour of data where the temperature was 24 or less and the air pressure (pressure\_kPa) was greater than 96.83. Provide an additional comment in your SQL script that states the number of rows returned. Execution: 1.5 mark Accuracy: 4 marks**

1000 rows returned.

**A screenshot of a computer

Description automatically generated**

1. **Select the date and time (date\_time), temperature (temp) and temperature flag (temp\_flag) columns for every hour where temperature data has been marked as missing using an ‘M’ value in the temp\_flag column. In the report, record the date and time. Hint: remember that single quotes are required for text values. Execution: 1.5 mark Accuracy: 2 marks**

We did not screenshot this answer of the question after we finished the whole 15 questions.

If you run this code for the first time, it should show 1 row returned, but if you run this code after executing all 15 questions, it should show 65 rows returned.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. **Return the same results as in Question 8, except construct the query to return both ‘M’ and ‘Missing’ values. Note: There are a number of ways of doing this, please use LIKE Execution: 1.5 mark Accuracy: 2 marks**

65 rows returned.

This outcome won’t be affected because it return both ‘M’ and ‘Missing ‘value.

**A screenshot of a computer

Description automatically generated**

1. **Count (SELECT COUNT) the number of hours contained in this dataset, where the dew point temperature (dew\_pt) is greater than one. Execution: 1.5-mark Accuracy: 2 mark**

6905 hours

**A screenshot of a computer

Description automatically generated**

1. **Count the number of hours that the wind was blowing (wind\_spd\_kmh is greater than 0) and from a direction ranging between north and east (wind\_dir\_10d is between 0 and 9) Execution: 1.5 mark Accuracy: 4 marks.**

7585 hours

**A screenshot of a computer

Description automatically generated**

1. **In the report only, in Barrie what percent of the time are the conditions listed in Q11 true? Written answer: 1 mark**

49.4523%

**A screenshot of a computer

Description automatically generated**

1. **Select all data for every hour of data where either the temperature (temp) is below freezing, or wind chill (wind\_chill) was below zero. Additionally, the wind speed flag (wind\_spd\_flag) must indicate that there were no problems with the wind speed instrument (i.e. no M or Missing values present) BTW: an anemometer is an instrument that measures wind speed Execution: 1.5 mark Accuracy: 6 marks**

5007 rows affected

SELECT \*FROM mydatabase.barrie\_weatherWHERE (temp < 0 OR wind\_chill < 0) AND wind\_spd\_flag NOT LIKE 'M' AND wind\_spd\_flag NOT LIKE 'Missing';

rows returned.

A screenshot of a computer

Description automatically generated

1. **Fix the M vs Missing problem in the temperature flag field by using UPDATE to change any instances of *Missing* values to just plain *M*. Execution: 1.5 mark Accuracy: 3 marks**

64 rows affected. You have to run the first line first, then run the second one.

SET SQL\_SAFE\_UPDATES = 0;

UPDATE mydatabase.barrie\_weather

SET temp\_flag = 'M'

WHERE temp\_flag = 'Missing';

A screenshot of a computer

Description automatically generated

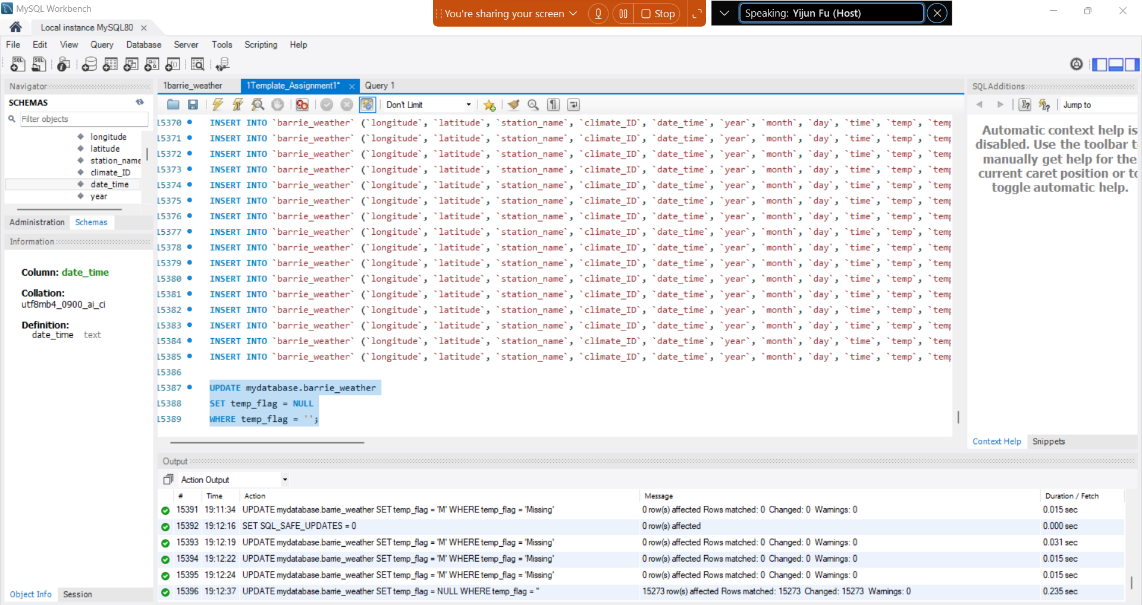
1. **Use an update statement to change all ' ' values in the temp\_flag column to NULL (no quotes, just the letters N-U-L-L) this will change. Hint: The values should end up looking like those in the humidex (hmdx) column. FYI: The best way to store null data is using the NULL value. Look to the humidex (hmdx) column for an example of this being implemented properly. Execution: 1.5 mark Accuracy: 3 marks**

15273 rows affected by the update.

UPDATE mydatabase.barrie\_weather

SET temp\_flag = NULL

WHERE temp\_flag = '';

****