Assignment 2 – User Defined Functions

Submission

Your submission will consist of two files:

- A single text-based SQL file
  – and –
- An output file demonstrating that your stored procedures work

Reminders

- 1 – Make sure you use proper indentations
- 2 – Make sure you fully comment your stored procedure
- 3 – Make sure you include error handling and test your error handling code

Tasks

1 – (0%) This is preparation – It is very important to ensure you have successfully ingested the data into both tables and have the exact number of rows specified below. Double check this.

- Create the EMPLOYEE table and STAFF table based on DDL in the dbs501-assignment-DDL file
- Take file dbs501-assignment-employee and load this data into an employee table
- Take file dbs501-assignment-staff and load this data into a staff table
- Make sure all records have been successfully loaded or your result sets may be incorrect
- Perform a SELECT COUNT(*) from both tables to ensure there is an exact match with rows

- You should have 35 records in STAFF and 42 records in EMPLOYEE

You can use INSERT, IMPORT, LOAD – whatever you prefer – to ingest the data

2 – (40%) Write a function called *my_median* which takes the values in a column (like salary) as input and will output the mathematical median of the values in that column.

The definition of median is:

X is the **ordered** list of elements
n is the number of elements in the list

Median(X) = X [(n+1)/2], if n is odd
Median(X) = (X [n/2] + X[(n/2)+1])/2, if n is event

Example:
Ordered list X = {5, 7, 10, 12, 15, 17}
Median(X) = ((X(3)+X(4))/2) = (10+12)/2 = 11

Ordered list X = {3, 5, 8, 15, 16}
Median(X) = X(3) = 8

Remember the list must be ordered, so, you must first ensure the list is ordered in your UDF. If the list input is:

X = {5, 2, 8, 1, 5} you must first order the list to X = {1, 2, 5, 5, 8}

WHAT TO HAND IN: A copy of your user defined function and the output of 3 calls showing your UDF works properly. Make sure to handle errors (like an empty list). Your 3 calls should be:

- A list with an event amount of elements
- A list with an odd amount of elements
- An empty list

3 – (40%) Write a function called *my_mode* which takes the values in a column (like department) as input and will output the mathematical mode of the values in that column.

The definition of mode is the value which occurs most frequently in the list. There are three cases:
A – There is one mode in the list
B – There are multiple modes in the list
C – There are no modes in the list

Example of A:
X = {1, 2, 3, 4, 2, 6, 5, 5, 2, 3, 2}
Mode = 2, since 2 occurs 4 times

Example of B:
X = {1, 2, 7, 4, 3, 2, 6, 5, 5, 8, 0}
Mode = 2, 5 – since both numbers occurs twice while all others are once

Example of C:
X = {1, 2, 3, 4, 5, 6, 7, 8. 9. 0}
There is no mode since all numbers occur exactly once

WHAT TO HAND IN: A copy of your user defined function and the output of 4 calls showing your UDF works properly. Make sure to handle errors (like an empty list). Your 4 calls should be:

- A list with zero modes

- A list with one mode
- A list with two modes
- An empty list

4 – (20%) – Write a procedure called *my_math_all* which takes the values in a column (like department or salary) as input and will output the mathematical median, mode and mean of the values in that column.

Median and mode are defined above.
Mean is the same as average and there's already a function called AVG in SQL which you can use.

WHAT TO HAND IN: A copy of your stored procedure and the output of 2 calls showing your stored procedure works properly. Make sure to handle errors (like an empty list). Your 2 calls should be:

- A list which shows a median, mode and mean (average)
- An empty list