Senior Data Engineer – Python assessment

Task

Develop a Python application that visualises the Share of search data for 3 terms: Football, Rugby and Tennis over the past 3 months.

To accomplish this please use Google Trends data as the data source. Additionally, we would like you to present this data using a suitable graph to effectively present this data. In case access to the Google Trends API is not feasible, explore alternative datasets consumable with Python for achieving the same objective.

DB Developer – SQL Assessment

Introduction

We process a large amount of web traffic data that is then used when we review the performance of our Direct Response advertising. For most clients we use our own tracking script which collects around 12M daily web page hit records across those clients. For a few of our other large clients we process their own sourced web visit data and some of the tables contain billions of web traffic records.

For the 1st interview stage you will need to solve the SQL assessment questions in this document: a strong, commercially applicable use of T-SQL is important for this role. Your answers should be sent to Client as soon as possible and will be used in a screening process for candidates to pass to the 2nd Interview stage, a video conference call with the Director of Technology.

Be prepared to answer questions about your assessment answers during that conference call interview. Ideally limited use of Google should be made during this 1st stage assessment. You will be asked if you needed to use Google.

Task 1: Find the clashes

For this task you need to write code to find the clashing records:

- *No Googling the specific scenario*
- *Please write the query with "best practice" code structure in mind (how you would do it in production)*
- *Try to not only solve this, but solve it with optimised code*
- *Consider how you would explain why you solved the problem with your specific method*
- *All code should be in T-SQL*

Consider that you want to identify clashes for a set of trainers based on their class schedule. Specifically, the class that starts while another class is still busy. Create a query to pull back all records that have overlapping datetimes for a specific Trainer from the sample rows below. In other words: clashes for each individual trainer, not clashes with another trainer's classes. Your query should identify the clashing records as per the sample, but also for any other rows that can be added into the table in the future). Ensure your solution also work for new data that might be added into the table.

Table A

trainerid	starttime	endtime
1234	01/10/2018 08:30	01/10/2018 09:00
1234	01/10/2018 08:45	01/10/2018 09:15
1234	01/10/2018 09:30	01/10/2018 10:00
2345	01/10/2018 08:45	01/10/2018 09:15
2345	01/10/2018 09:30	01/10/2018 10:00
2345	01/10/2018 10:50	01/10/2018 11:00
2345	01/10/2018 09:50	01/10/2018 10:00

For this dataset above your query should identify that there are clashes for the following records only:

trainerid	starttime	endtime
1234	01/10/2018 08:45	01/10/2018 09:15
2345	01/10/2018 09:50	01/10/2018 10:00