Data Engineer Job Technical Test

Numerical Analysis

Pens Inc is a startup that manufactures and sells pens.

Pen Manufacture

Pens Inc have 2 machines which can each produce 10 pens per hour. Production runs for 6 hours per day, 5 days a week. Pens are kept in storage until they are sold on the website. The storage can hold a maximum of 2000 pens. Pens Inc begins manufacturing pens for the very first time at the start of week 1, and the storage is empty.

a. Pens Inc have launched their first product, which sells 100 units in the first week, 120 in the 2nd week and 144 in the 3rd week. If sales continue to grow at this rate, how many pens will they have sold in total 8 weeks after launch?

b. The storage is to be emptied every 8 weeks. What is the minimum number of pens that need to be sold per week to be sure that the storage doesn't exceed maximum capacity by the end of week 8?

Pen Sales

Pens Inc sells its pens on its website.

- The pens retail for £10/unit and cost a total of £6/unit to manufacture
- The only traffic to the site comes via Cost per Click (CPC) traffic - there is no organic traffic. The conversion rate on that traffic is 20%

1. What is the maximum CPC they should be willing to pay to ensure they break even?
2. What would that maximum CPC become if the conversion rate changed to 30%?
3. Every week, Pens Inc spends £300 on digital marketing and sells 200 pens. One week, they decided to increase spending to £350 and sell 210 pens. What is the cost per incremental sale in that week?

International Expansion

Pens Inc expands and begins selling its pens in France & Switzerland. They are using the following currency conversions:
1 GBP = 1.132 EUR
1 EUR = 1.149 CHF

They sell pens in France for 12 EUR each and in Switzerland for 15 CHF each. The pens are still made in the UK at the same cost as before. In one week they sell 200 pens in France and 300 in Switzerland.

1. What is the total profit, in GBP, made on these sales, ignoring shipping and FX fees?

SQL Analysis

Pens Inc grows considerably in 3 years, and now maintains a database to store its orders. Their relational database contains the following four tables:

users (
  user_id **INT PRIMARY KEY**
, first_name **VARCHAR**(255)
, last_name **VARCHAR**(255)
, date_created **DATE**
, email **VARCHAR**(255)
)
	orders (
	  order_id **INT PRIMARY KEY**
	, user_id **INT**
	, date_created **DATE**
	, order_value **FLOAT**
	, city_id **INT**
	)

city (
  city_id **INT PRIMARY KEY**
, city_name **VARCHAR**(255)
, country_id **INT**
)
	country (
	  country_id **INT PRIMARY KEY**
	, country_name **VARCHAR**(255)
	, currency_code **VARCHAR**(255)
	)

Please create an executable SQL statement to answer the questions below.

Note that you will be marked on whether the statement will run but also whether your statement is optimized and set out in a readable format.

Please indicate which SQL dialect you are using, e.g. “MySQL, PostgreSQL, Oracle…”

1. Count the number of customers that placed at least 1 order, whose last_name is “Scott”.

2. Count the number of different cities where at least 1 order has been placed yesterday (do not hardcode the date).

3. Find the weekly order count for the city of Geneva for the last 8 weeks, and also the cumulative total.Desired output: [week_start, order_count, cuml_order_count]

4. Find the distribution of first-order values per user in the UK, using a bin width of £10 and a maximum order value of £50. You can not assume that the order ID field is ordered logically or sequentially. Desired output: [order_value_bucket, order_count]

5. For each user output the mean and maximum number of days between two consecutive transactions. Return "-1" for users with only one transaction; e.g. if a user had transactions on ‘2019-01-01’, ‘2019-01-02’ and ‘2019-01-05’ then the number of days between transactions is 1 and 3, the average being (1+3)/2 = 2 and maximum being 3.

6. Assuming that you have a query that runs very slowly, what are the steps that you would perform in order to identify the cause?

Python Questions

1. How would you remove duplicates within a list in Python?

2. Given a list of integers, and an integer N, write a function “sum_to_n” to find all combinations that sum to the value N.

    Input:

    integers = [2,3,5]
    N = 8

    Output

    def sum_to_n(integers, N) ->
     [
      [2,2,2,2],
      [2,3,3],
      [3,5]
     ]

3. Given two string arrays word1 and word2, return true if the two arrays represent the same string, and false otherwise.
A string is represented by an array if the array elements are concatenated in order to form the string.

Example :
Input: word1 = ["ab", "c"], word2 = ["a", "bc"]
Output: true
Input: word1 = ["ab", "c"], word2 = ["ac", "b"]
Output: false