Technical Assessment for Data Engineering Role

Dear Candidate,

Thank you for expressing your interest in joining Datumlabs. We are excited to embark on this journey with you. This document marks the beginning of our interview process, designed to explore your skills and provide a glimpse into the types of challenges you'll encounter in your day-to-day work with us.

Contained within are a series of interview questions covering Python and SQL. These have been carefully crafted to assess your problem-solving abilities and technical expertise. At the end of this document, you will find sample tables, complete with schema and data, which will be essential for your SQL tasks.

We encourage you to tackle as many questions as you can. When formulating your solutions, please include clear explanations and sufficient documentation to convey your thought process and any alternative approaches you considered.

Upon commencing work on these questions, kindly inform us. You will have one week from the start date to complete and submit your solutions. Should you encounter any difficulties or have questions about the tasks, please feel free to reach out to us without hesitation.

We look forward to reviewing your responses and wish you the very best in this assessment phase. Your success and progress are important to us, and we are here to support you throughout this process.

Best regards,

The Datumlabs Team

Python Questions

1. Fill None Values: Given a list, replace None values with the previous non-None value. If consecutive Nones occur, fill each with the last non-None value. Example: [1, None, 1, 2, None] becomes [1, 1, 1, 2, 2].

2. Mismatched Words Finder: Write a function that returns a list of words present in two strings that don't match in case. Example: Input: "Datumlabs is an awesome place", "Datumlabs.io Is an AWESOME place".
   Output: ["is", "Is", "awesome", "AWESOME"].

3. Character Frequency Counter: Create a function to count the occurrences of a specific character in a string.
   Example: 'mississippi', 's' should return 3.

4. Nth Largest Value Key Finder: Write a function to find the key of the nth largest value in a dictionary. Example:
   For {a: 1, b: 2, c: 100, d: 30}, and n = 2, return 'd'.

SQL Questions

1. Percentage of Paid Customers Who Bought Both Product A and Product B: Given a table CustomerPurchases with columns customer_id, product_id, purchase_date, price, and payment_status, calculate the percentage of customers who bought both productsA and B and paid for them.

2. Percentage of Sales Attributed to Promotions on First and Last Days: With the Sales table (columns:
   sale_id, product_id, sale_date, amount, promotion_id) and Promotions table (columns: promotion_id, start_date, end_date, discount_rate), compute the percentage of sales attributed to promotions on their first and last days.

3. Top 5 Complementary Products for Product A: Identify the top 5 products bought alongside Product A.

DBT/Pyspark Metrics Calculation:

Using dbt or PySpark (as per your convinience), compute the following metrics with provided sample data tables (UserActivity, Users, Sales, Products, Categories):

1. Monthly Active Users (MAU) for January 2024: Count of unique users active in January 2024.
2. Total Sales Revenue for January 2024: Sum of sales in January 2024.
3. Average Sale Amount Per Category for January 2024:Average sale amount per category in January 2024.
4. Number of New Users in January 2024: Count of users who joined in January 2024.
5. Top Selling Product Category in January 2024: Product category with highest sales in January 2024.

Sample Table: CustomerPurchases

Column	Data Type	Description
customer_id	VARCHAR	Unique identifier for the customer
product_id	VARCHAR	Unique identifier for the product
purchase_date	DATE	Date of purchase
price	DECIMAL	Price of the product
payment_status	VARCHAR	Status of payment (e.g., 'paid')

customer_id	product_id	purchase_date	price	payment_status
C001	A	2024-01-01	50.00	Paid

C001	B	2024-01-05	30.00	Paid
C002	A	2024-01-10	50.00	Paid
C003	C	2024-01-15	20.00	Paid
C002	B	2024-01-20	30.00	Unpaid
C004	A	2024-01-25	50.00	Paid
C004	B	2024-01-30	30.00	Paid

Sales

Column	Data Type	Description
sale_id	VARCHAR	Unique identifier for the sale
product_id	VARCHAR	Unique identifier for the product
sale_date	DATE	Date of sale
amount	DECIMAL	Amount of sale
promotion_id	VARCHAR	Identifier for any promotion applied

Sales

sale_id	product_id	sale_date	amount	promotion_id
S001	A	2024-01-01	45.00	P001

S002	B	2024-01-02	25.00	P002
S003	A	2024-01-03	50.00	None
S004	C	2024-01-04	18.00	P001
S005	B	2024-01-05	30.00	None

Promotions

Column	Data Type	Description
promotion_id	VARCHAR	Unique identifier for the promotion
start_date	DATE	Start date of the promotion
end_date	DATE	End date of the promotion
discount_rate	DECIMAL	Discount rate of the promotion

Promotions

promotion_id	start_date	end_date	discount_rate
P001	2024-01-01	2024-01-07	10%
P002	2024-01-02	2024-01-08	15%

Sample Table: UserActivity

activity_id	user_id	activity_date
1	101	2024-01-05
2	102	2024-01-06
3	103	2024-01-07
4	101	2024-01-15
5	104	2024-01-20
6	102	2024-01-25
7	105	2024-01-30

Sample Table: Users

user_id	user_name	join_date

101	Alice	2023-05-10
102	Bob	2023-06-15
103	Charlie	2023-07-20
104	Dana	2023-08-25
105	Emily	2023-09-30

Sample Table: Sales

sale_id	product_id	sale_date	amount	category_id
1	P001	2024-01-01	100.00	C1
2	P002	2024-01-05	150.00	C2
3	P001	2024-01-10	100.00	C1
4	P003	2024-01-15	200.00	C3
5	P002	2024-01-20	150.00	C2

Sample Table: Products

product_id	product_name	category_id
P001	Product A	C1
P002	Product B	C2
P003	Product C	C3

Sample Table: Categories

category_id	category_name
C1	Electronics
C2	Clothing
C3	Home Appliances