Data Engineer Part 1: SQL-Focused Problems

Task: You are building a database for a PowerToFly website to show users' skills. Each user can have multiple skills (Python, Java, SQL, NodeJS, etc…). You need to design the database schema and write a SQL query to find users that have at least 5 of the top 10 skills.

Questions:

- Can you explain the performance of your query?
- How can you improve the query performance?
- Can you suggest different indexes or any other ways to optimize the query execution time?

An example task that we can give to candidates to assess their ability to write queries using window functions (PostgreSQL if possible)

Task: You have a table that contains information about user work experience. The table has the following columns:

- id (integer)
- user_id (integer)
- title (text)
- description (text)
- date_added (date)
- date_edited (date)

You need to write a SQL query to get only 2nd work experience, for example

user King Arthur has experience:

id	user_id	title	description
1000	1	Farmer	Collected potatoes
1001	1	Knight	Killed enemies
1002	1	King	Ruled the kingdom

The result of the query should be a record with id == 1001

Questions:

- Can you explain the performance of your query?
- How can you improve the query performance?
- Can you suggest any other approach to solve this problem?

Part 2: PySpark Test

Environment: Databricks or EMR cluster

Requirements: PySpark, Delta Lake, AWS S3

QUESTIONS: Analyzing Customer Reviews (40 minutes)

You have been provided with a large dataset of customer reviews in Delta Lake format on AWS S3: s3://your-bucket/customer_reviews/

The dataset contains a mixture of structured and unstructured data. Each line in the dataset represents a customer review, and has the following format in a set of delta files:
product_id, user_id, user_name, rating, review_date, review_title, review_text

Your task is to write a PySpark script that reads this dataset and performs the following analysis:

However, due to the presence of messy data, some lines may contain extra or missing fields, improperly formatted dates, or special characters.

1. Create a table (format choice is yours) with the following columns:

   uuid (String): Unique identifier for each record, generated based on product_id, user_id, and the Unix timestamp of the review_date
   product_id (String): Unique identifier for each product
   user_id (String): Unique identifier for each user
   user_name (String): Name of the user
   rating (Float): Rating given by the user, ranging from 1.0 to 5.0
   review_date (Date): Date of the review submission
   review_title (String): Title of the review
   review_text (String): Text of the review

2. Read the dataset and validate the data by:
   Validating each column using regex patterns, as applicable
   Removing any special characters from the user_name, review_title, and review_text columns
   Filtering out rows with an incorrect number of fields. If missing more than 40% of data.
   Converting improperly formatted dates in the review_date column to a standard date format (e.g., "yyyy-MM-dd")
   Alerting if a row does not match the regex patterns or contains an improperly formatted date (only if you dont clean the data col)
   Remove columns if column is missing more than 90% of data.

3. Calculate the average rating for each product.

4. Find the top 10 products with the highest average ratings.

5. Identify the top three users with the most reviews submitted.

Please use appropriate PySpark APIs and follow best practices for code readability and optimization. Ensure that your solution is optimized for processing large datasets (100 TBs) by leveraging big data best practices.