Case Study Assignment for Data Engineer

The case study consists of two parts:

- **Hands-on Assignment** – tests the technical abilities of a candidate in an example that is very similar to a daily bread of a data engineer with focus on relevant technology, concepts, and programming languages.
- **Open Questions** – tests conceptual understanding of a candidate about expanding the developed solution further.

Please don’t forget to collect the required output documents and send them to a recruiter organizing the interview.

Feel free to use AI tools to help you with your assignment, make sure to validate the solution works and you can interpret it well.

You can contact petr.svec@merkle.com with any questions related to the assignments.

Hands-on Assignment

Prerequisites

- Identify files needed for the exercise saved in the public S3 bucket:
    - https://merkle-de-interview-case-study.s3.eu-central-1.amazonaws.com/de/item.csv
    - https://merkle-de-interview-case-study.s3.eu-central-1.amazonaws.com/de/event.csv
- Spin up a free Databricks “for personal use” instance available after login at https://databricks.com/try-databricks.

Assignment

- Load prerequisite files from S3 to HDFS (or alternative like DBFS or S3).
- Using your business acumen understand the content of the files.
- Create a spark script to create a small data lake that would consist of:
    - 1. Layer
        - Contains external tables for all prerequisite files.
        - All attributes are of STRING type.
        - No transformations are applied.
    - 2. Layer
        - Contains all datasets from the first layer.
        - All attributes have common naming convention.
        - All attributes have proper datatypes based on the attribute name and common logic.
        - All struct collection attributes are flattened and transformed to proper data types.
        - Fact tables are properly partitioned based on meaningful attributes.
    - 3. Layer
        - “**top_item**” datamart
            - The datamart shows for every item and year:
                - Total number of item views in a particular year.
                - Rank of an item based on number of views in a particular year.

        - The most used platform for particular item in particular year.

Open Questions

With regards to the solution from the hands-on assignment, please answer the following questions:

- What steps are missing to industrialize such solution further.
- If the solution was implemented in dbt-core, how would the overall architecture change? Would there be another cloud resources needed?
- What would implementation in dbt-core bring to the project. What would be the upsides and downsides.
- Please estimate the effort you'd requested to implement the solution in dbt-core.

Expected Output

- Public Git repository, such as GitHub with:
    - **IPython Notebook** with Python script that contains all actions necessary to fulfil the assignment including required DDLs
        - All code blocks should be documented
        - Data mart creation and attributes are documented using Markdown
    - Markdown file with answered open questions.