Data engineer challenge

Context

Your company provides a feature that allows users to schedule recurring allowances. This feature enables users to set a specific amount and periodicity (e.g., daily, weekly, biweekly, or monthly) for payments they will receive.

The backend process responsible for updating the system's allowance and payment schedule tables experienced issues, resulting in discrepancies specifically in the **next payment day** fields across the datasets. The data you are provided reflects all recorded events and backend table states **up to December 3, 2024**, which you should consider as the **current day**.

The backend tables operate as follows:

1. **Allowance Backend Table (`allowance_backend_table`)**
    - **Purpose**: Stores the current allowance settings for each user, reflecting their most recent allowance configuration.
    - **Operation**:
        - When a user **creates** or **edits** an allowance, this table should be updated to reflect the latest `frequency`, `day`, and `next_payment_day`.
        - The **`next_payment_day`** field should accurately represent the upcoming payment date based on the user's allowance settings.
        - The allowances get created with the `status` column `enabled`, but the users can turn them to `disabled`. Make sure you only use the enabled allowances.
    - **Potential Issues**:
        - Discrepancies during the affected period may have resulted in incorrect `next_payment_day` values in this table.
2. **Payment Schedule Backend Table (`payment_schedule_backend_table`)**
    - **Purpose**: Manages the scheduling of payments to users based on their allowance settings.
    - **Operation**:
        - Each user should only have **one active record** in this table at any time.
        - When a payment is made:
            - The current record for the user is **deleted**.
            - A new record is **created** with the upcoming `payment_date`.
        - The **`payment_date`** should align with the `next_payment_day` from the `allowance_backend_table`.
        - Users who disable the allowance should **not** have any active record.
    - **Potential Issues**:

        - Errors during the affected period may have resulted in users having **multiple records** or incorrect `payment_date` values in this table.

Importantly, the logs of events related to user actions—such as creating or editing allowances—are believed to be accurate and can be considered the **source of truth**.

Your task is to analyze the provided datasets and create a detailed report describing any discrepancies or patterns you observe. The goal is to guide the backend team by providing a thorough understanding of what went wrong.

Your Goal

Using the `allowance_events` dataset as the **source of truth**, analyze the data and create a detailed report describing any discrepancies or patterns you observe in the **`next_payment_day`** and **`payment_date`** fields across the backend tables. Be as explicit and thorough as possible in your findings to help the backend team understand the scope of the problem and what might have gone wrong.

Available Data

You have access to the following datasets:

1. **`allowance_events`** (JSON)
   This dataset captures the creation and updates of user allowances during the affected period.
   - **Fields**:
     - `event.name`: The type of event, either `allowance.created` or `allowance.edited`.
     - `event.timestamp`: Timestamp of the event.
     - `user.id`: Unique identifier for the user.
     - `allowance.amount`: Allowance amount.
     - `allowance.scheduled.frequency`: Frequency of the allowance: `daily`, `weekly`, `biweekly`, or `monthly`.
     - `allowance.scheduled.day`:
       - `"daily"` for daily frequency.
       - Day of the week for weekly/biweekly schedules.
       - `"1st"` or `"15th"` for monthly schedules.
2. **`allowance_backend_table`** (CSV)
   This table contains backend records of allowances during the affected period. It has

been observed that the **`next_payment_day`** field in this table may contain errors or inconsistencies.
    - **Fields**:
        - `uuid`: Unique identifier for the user (corresponds to `user.id` in events).
        - `creation_date`: Timestamp when the record was created.
        - `frequency`: Allowance frequency (`daily`, `weekly`, `biweekly`, `monthly`).
        - `day`: Scheduled day for the allowance.
        - `updated_at`: Timestamp of the most recent update to the record.
        - **`next_payment_day`**: The next scheduled payment day as a float (1 to 31). This field is suspected to contain discrepancies.
        - `status`: indicates if the allowance is currently `enabled` or `disabled`
3. **`payment_schedule_backend_table`** (CSV)
   This table contains the payment schedule records. It has been observed that the **`payment_date`** field may contain errors or inconsistencies.
    - **Fields**:
        - `user_id`: Unique identifier for the user (corresponds to `user.id` in events).
        - **`payment_date`**: Scheduled payment day. This field is suspected to contain discrepancies.

Key Evaluation Points

We will evaluate your submission based on two main aspects:

1. **Quality of the Report**:
    - Clarity and depth in describing the findings.
    - Thoroughness and accuracy in identifying discrepancies and patterns.
    - Logical reasoning and structure in presenting your observations.
2. **Quality of the Code**:
    - Cleanliness, organization, and documentation of the code.
    - Correctness and efficiency in implementing the analysis.
    - Appropriate use of tools and methods to process the data.
    - Ideally, the code should be shared via a link to a repository on GitHub or a similar platform, allowing us to review your work in a structured and collaborative environment.