Data Maturity SCD

Naam: Mohamed Ajimi
Datum: 25/04/2024

1. Inhoud

2. Introduction

In the context of a marketing-driven questionnaire project, our objective is to implement an SCD Type 2 strategy within Google BigQuery to track and manage changes to questions, answers, scores, and hierarchical categorizations (categories and subcategories). This strategy ensures that any modifications made by the marketing team—such as adding new questions, changing answer options, or adjusting scoring algorithms—are captured as new versions in the data warehouse without overwriting historical data. This capability is crucial for analyzing trends over time and assessing the impact of changes on user responses.

To facilitate these changes, we employ several tools and platforms:

- **Google Forms** serves as the primary interface for capturing user responses.
- **Google Sheets** acts as the master database for questions, answers, and other metadata, providing a familiar and accessible interface for making changes.
- **AppSheet** is integrated with Google Sheets to offer a user-friendly platform for the marketing team to make changes directly, without needing direct access to the database or technical tools.
- **Google Cloud Functions** are used to automate the ingestion of updates from AppSheet into BigQuery, handling both new data entries and updates to existing data.
- **Google BigQuery** stores all historical and current data, applying the SCD Type 2 model to track the lifecycle of each piece of data.

The integration of these tools allows for a streamlined process where changes made in AppSheet are automatically reflected in BigQuery, maintaining a clear and accurate historical record of every change. This system supports advanced data analysis capabilities, enabling the marketing team to see how changes affect user engagement and response patterns over time.

3. Tools:

3.1. Central Role of Google Sheets

Google Sheets serves a dual purpose in our project: as a central repository for all survey-related data and as a structured mirror of the BigQuery tables, facilitating seamless updates and integrations. This setup enhances the manageability and accessibility of data changes, crucial for the dynamic nature of our marketing-driven questionnaire project.

- **Table Alignment:** The structure of the sheet directly corresponds to the tables in BigQuery, such as `D_Question`, `D_Answer`, `D_SubCategory`, `D_Category`, and `D_QuestionType`. This alignment ensures that any changes made in the sheets can be directly and seamlessly mapped to the corresponding BigQuery tables.
- **Ease of Integration:** By mirroring the structure of the BigQuery tables, we simplify the integration process between Google Sheets and BigQuery via AppSheet and Google Cloud Functions. This setup minimizes the risk of errors during data transfer and streamlines the update process.
- **Dynamic Interface Creation:** AppSheet uses the structured data in Google Sheets to dynamically create a user interface that allows marketing team members to make real-time changes to the questionnaire.
- **Direct Updates:** Changes made in AppSheet are written back to Google Sheets, ensuring that the central repository is always up to date.

3.2. AppSheet: Dynamic Data Management Interface

- **User Interface and Data Entry:** AppSheet serves as the dynamic front-end for our data management system. It allows marketing team members, regardless of their technical expertise, to enter and update data related to the data maturity project directly. The interface is customized to reflect the structure and fields present in Google Sheets, ensuring data consistency.

- **Real-Time Data Interaction:** AppSheet provides a live link with Google Sheets, enabling immediate reflection of data changes. This capability allows for the real-time application of modifications without manual intervention, streamlining data management tasks.

3.3. Google Cloud Functions: Automating Data Processes

Data Ingestion and Management: Google Cloud Functions play a crucial role in automating the ingestion and management of data updates from AppSheet into Google BigQuery. These functions handle the logic for data transformations, applying business rules, and ensuring that data updates follow the SCD Type 2 model.

Workflow Automation: Through the use of webhooks and triggers set up in AppSheet, Google Cloud Functions are automatically called upon data changes. This automation significantly reduces manual overhead and speeds up the data processing cycle, ensuring timely updates to the data warehouse.

4. Implementation Steps

This section describes the step-by-step process by which data is captured, processed, and stored in our data management system. It highlights the flow from AppSheet, through Google Sheets and Google Cloud Functions, to Google BigQuery.

4.1. Step 1: Data Capture via AppSheet

User Interface Interaction: Marketing team members use the customized interface on AppSheet to enter new data or update existing data. This interface is directly linked to Google Sheets, ensuring that data entries are structured according to the existing database schema.

Data Validation: AppSheet provides built-in validation rules which ensure that all data entries meet the required standards before submission. This prevents errors and inconsistencies in data at the source.

4.2. Step 2: Synchronization with Google Sheets

Immediate Data Sync: Once data is entered or updated in AppSheet, it is immediately synchronized with Google Sheets. This synchronization ensures that Google Sheets always holds the latest data as the central repository.

Structured Data Storage: Google Sheets maintains a structured format that mirrors the BigQuery tables, which is crucial for maintaining data integrity and streamlining subsequent data processing steps.

4.3. Step 3: Data Processing by Google Cloud Functions

In this critical step, automation bots in AppSheet detect changes in Google Sheets and trigger Google Cloud Functions. These functions are designed to manage and process data updates through a series of automated, reliable, and scalable steps.

1. Automation Bots

Addition Bot

Functionality: These bots are specifically designed to monitor the addition of new data entries in Google Sheets. Whenever a new row is added to tables such as `D_Question`, `D_Answer`, `D_SubCategory`, `D_Category`, or `D_QuestionType`, the Addition Bots trigger a webhook.

Role: The webhook initiated by this bot calls upon a designated Google Cloud Function that handles the insertion of new data into BigQuery, ensuring that each new entry is processed according to the SCD Type 2 model, thereby maintaining historical data integrity.

Change Detection Bot

Functionality: These bots are responsible for identifying changes made to existing rows in any of the designated tables in Google Sheets. It monitors for any updates, modifications, or deletions made to the data.

Role: Upon detecting a change, the Change Detection Bot sends a webhook to a specific Google Cloud Function that is tasked with handling these updates. This includes executing the SCD Type 2 strategy where updates result in new records with updated version flags and timestamps, preserving the historical versions of the data.

2. Google Cloud Functions

Data Transformation and Loading

SCD_function: This function is crucial for adding new data rows. It processes the data sent from the Change Detection Bots.

update_delete_SDC_function: Responsible for handling updates and deletions in the data, this function works closely with the SCD_function. For deletions, it manages how these are recorded, ensuring that the historical integrity of the data is not compromised by physically removing any data.

☐	✓	2nd gen	SCD_gold_function	Apr 11, 2024, 9:12:51 AM	europe-west1	HTTP	Python 3.12	256 MiB	add_or_update_data_gold
☐	✓	2nd gen	SDC_function	Apr 11, 2024, 9:15:31 AM	europe-west1	HTTP	Python 3.12	256 MiB	add_data
☐	✓	2nd gen	update_delete_SDC_function	Apr 14, 2024, 10:31:03 PM	europe-west1	HTTP	Python 3.12	256 MiB	process_scd_type_2
☐	✓	2nd gen	update_linked_answers_function	Apr 14, 2024, 11:12:36 PM	europe-west1	HTTP	Python 3.12	256 MiB	update_answers

Interactions between functions and bots

- The Google Cloud Functions are tightly integrated with the automation bots. When a bot detects a change or addition, it not only triggers the appropriate function but also passes along the necessary data payload. This payload includes details of the changes along with metadata such as timestamps and user IDs when applicable.

```
{
  "Timestamp": "<<NOW()>>",
  "UserEmail": "<<USEREMAIL()>>",
  "TableName": "D_Question",
  "ActionType": "Add",
  "RowData": {
    "questionID": "<<[questionID]>>",
    "questionTypeID": "<<[questionTypeID]>>",
    "subCategoryID": "<<[subcategoryID]>>",
    "questionDescr": "<<[questionDescr]>>",
    "categoryID": "<<[categoryID]>>",
    "maximumScore": "<<[maximumScore]>>",
    "weight": "<<[weight]>>",
"current_state":"<<[current_state]>>",
"finalCalculatedWeight": "<<[finalweightvirtual]>>",
"subCatWeight": "<<[subcatvirtual]>>"
  }
}
```

3. Step 4: Storage and Version Control in Google BigQuery

Data Ingestion: Transformed data is ingested into Google BigQuery, where it is stored both historically and currently. This allows for efficient data retrieval and analysis.

☐	start_date	DATE	NULLABLE	-
☐	end_date	DATE	NULLABLE	-
☐	current_state	BOOLEAN	NULLABLE	-

Implementation of SCD Type 2: BigQuery implements SCD Type 2 to maintain historical versions of data alongside current versions. This is crucial for analyzing how changes over questionnaire versions affect user responses and engagement.

Query and Analysis: With data stored and structured appropriately in BigQuery, it can be queried to generate insights, reports, and dashboards. These analytics help the marketing team to evaluate the effectiveness of different questionnaire strategies and make informed decisions.