**Section 2: Marketing data modeling:**

**Handling hourly vs daily updates:**

The model handles daily and hourly updates by storing frequently updated data, such as bids, in the fact table (**fact\_marketing**\_events). This data can be updated more frequently than other data. For example, according to the DAG model scheduling, new records with updated data can be added to the fact table every hour or every day, while historical data is stored in the dimension tables in a different DAG model, which allowing for daily or high-frequency updates while maintaining a complete history and enabling up-to-date analysis.

**Tracking Historical Changes:**

1. **Dimension Tables:** The data in the dimension tables is stored as fixed fields that do not change frequently. When there are significant changes, for example, in the structure of the advertisement, campaign, or account, the records in these dimension tables can be updated while preserving history by using slowly changing dimensions tables.
2. **Fact Table:** The fact table contains date fields and events, allowing for tracking changes and performance analysis over time.

**Addressing business requirements:**

1. **Hierarchy -** The model accounts for the hierarchy between different entities, such as Account, Sub-Account, Portfolio, Campaign, Ad Group, and Ads.
2. **Historical Analysis and Real-Time Reporting -** The use of date fields and events in the fact table enables easy historical analysis and real-time reporting over time.
3. **Frequent Changes -** Fields such as **bid** and **Label** are stored in the fact table, which allows for frequent updates (hourly or daily scheduling of the DAG model).

**Trade-offs considered and approach:**

One of the main goals of the model was to enable efficient analysis and reporting for the marketing team. One of the best ways to approach this goal is to use star schema design with a large fact table. Although using a fact table can lead to a very large amount of data, but this decision was made as it allows for flexible queries and fast analysis.

**Handling data quality issues:**

1. Data definitions and constraints, such as **NOT NULL** and **DEFAULT**, ensure data consistency.
2. Validation and control processes can be implemented in production to ensure that all data entering the dimension and fact tables is accurate and consistent.
3. All columns that are subject to change frequently are stored in the fact table, which allows to handle frequent updates.

**Recommendations for indexing strategy to optimize query performance:**

1. Indexing in dim tables – creating indexes on primary key columns in dim tables to accelerate queries that perform joins between tables.
2. Indexing in fact table – creating indexes on the foreign keys columns in fact table to accelerate queries that perform joins with that table as well.
3. Partitioning – dividing the fact table into partitions according to dates to improve query performance when focusing on specific time ranges.