Data Engineer Take Home

Expected Duration

This task should take 3-6h. You have two business days from the time you receive this document to submit your response.

Task Description

You are given an SQLite db file which contains three tables. Your goal is to integrate data across the three tables:

1. users
2. conversation_starts
3. conversation_parts

You are to create a **consolidated_messages** table from these three tables. The end data model is tracking consolidated conversation between users. If you would create a different model than the example below, provide an explanation of what you would change and why.

For the **consolidated_messages** table, create three fact/dim schemata:

1. fact table schema
2. user dimension table schema
3. conversation_parts dimension table schema.

There is a **user** table with basic user information such as:

1. **id** - The id of the user
2. **email** - The user's email
3. **name** - The user's name
4. **is_customer** - Whether the user is a customer

There is a **conversation_start** table which contains the first message in the conversation and the following key fields:

1. **id** - the id of the conversation
2. **priority** - the importance of the message
3. **conv_dataset_email** - the email of the entity that started the conversation.
4. **message -** The content of the message
5. **created_at** - The time the message was sent

There is a **conversation_part** table which contains:

1. **id** - The id of the conversation part
2. **conversation_id** - The id of the conversation that the part belongs to
3. **conv_dataset_email** - The email of the user that was the author of the part
4. **part_type** - The type of message as an enum (comment, assign, close etc).
5. **message** - The content of the message
6. **created_at** - The time the message was sent

From these tables we want a resultant **consolidated_messages** table the combines all messages and user info, with the following fields:

1. **id** - The primary key of rows in this table
2. **user_id** - The id of the user that is the customer in the conversation (it is not necessarily the user that starts the conversation)
3. **email** - The email of the entity that sent the message
4. **conversation_id** - The id of the conversation that the message belongs to
5. **message** - The message that was communicated
6. **message_type** - A controlled vocabulary derived *mostly* from the **part_type** column in the conversation_part table.
7. **created_at** - The time the message was sent

Notes

- The message_type "**open**" is not explicitly in any of the tables. It should be inferred from the **conversation_start** table as that is the message that opens the conversation.
- Not all entities that participate in the conversations are customers.
- For each **consolidated_messages** row, the user_id value should contain the id of the user that is the customer.

Before committing to the new table, sort the **consolidated_messages** data by:

1. conversation_id
2. created_at

Data

1. The data will be provided to you as an SQLite DB file.
2. Create a github repo.
3. In python, use the sqlite library to access the db
4. Perform the integration, ideally using a single SQL query.
5. Store the result in a new table called conslidated_messages
6. Provide the 3 schema as SQL (see example below)
7. Provide an explanation for your SQL integration code.

8. Share the repo and the file when you are done.

Example

An example dimension schema definition might be:

```
dim_dates (Date Dimension)

sql
CREATE TABLE dim_dates
  ( date_id INT PRIMARY KEY,
    -- YYYYMMDD format date DATE,
    day INT,
    month INT,
    year INT );
```

For the **consolidated_messages** table, see the attached **consolidated_messages_example.csv** for a truncated expected result.

Evaluation Criteria

- Functionality
- Efficiency
- Reusability
- Style + Adherence to convention
- Documentation
- Communication