Table of Contents

Executive Summary

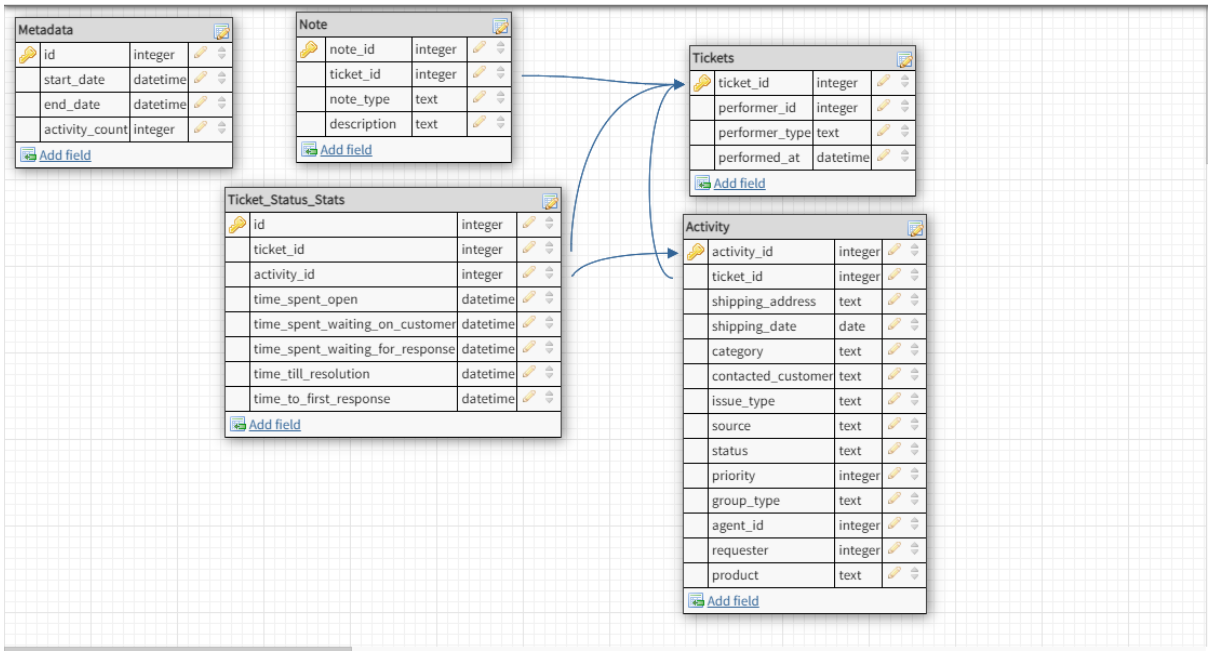
This report is based on programming task performed to generate random customer ticket data using Python v3.8.2 and SQLite. The report summarizes assumptions made around sample data, how to get started with the execution, setup, working of each program file, database design.

Assumptions

According to the sample json data given in the task instructions below assumptions around the data were made.

1. Metadata describes start date and end date between which all the tickets were created and their total count.
2. Activities_data contains list of all the tickets.
3. Each ticket in the activities_data has ticket_id which is unique, performed_at is a date of ticket creation, performer_id is id of user who created the ticket, activity contains details of the ticket.
4. Activity in each ticket can take 2 forms based in sample json:
   1. Note with id, type and description
   2. Product shipping details

Database Design



Working of Program Files

Create_Random_Tickets_Data.py

This python script generates json with random ticket data based on user's input. The script saves json file in the same directory.

Create_Tables.py

This python script reads the json generated by above script, processes it and converts the json into separate dataframes corresponding to each table to be created in sqlite.

df_metadata for Metadata table

df_tickets for Tickets table

df_activity for Activity table

df_note for Note table

This python script also creates database, tables and inserts corresponding data into tables in sqlite. The python library used here is sqlite3.

Generate_Stats.sql

This is sql script to create a new table Ticket_Status_Stats which will store time for which each ticket was in particular status –

Time spent Open = Open
Time spent Waiting on Customer = Waiting for Customer OR Waiting for Third Party
Time spent waiting for response = Pending Status
Time till resolution = Resolved
Time to first response = Closed

The sql script also creates a TRIGGER named 'UPDATE_STATUS_TIME_TRIGGER' on Activity table to update time in Ticket_Status_Stats table when status of ticket in Activity table changes. This trigger helps to automatically record the time of previous status whenever it changes to new one.

Getting Started

Step 1: Download the Sqlite (including CLI version) in your machine using below link. Make sure you add path to sqlite folder in ENVIRONMENT variables of your machine. To view tables in database you can also download DB Browser for Sqlite which provides GUI platform.

URL: https://www.sqlite.org/download.html

Step 2: Download Python v3.8.2 using below link and install it. Make sure you add path to python binaries in ENVIRONMENT variables of your machine.

URL: https://www.python.org/downloads/

Step 3: Clone the repository from github using following url. The repository is currently public.

URL: https://github.com/Ninad291994/Test.git

Step 4: Follow the instructions in README.MD file in the repository to execute the code.