Amanda Lian Huijie, A0221973A
Xu Zeng, A0194487L

Sie Xiang Yi, Jefferson, A0166887E
Kenny Chew, A0200016H

Group 11: Fake but Realistic Data

Overview

In our project, we will be implementing a Python Tkinter program that allows a user to generate customized mock data that is realistic. The program will consist of text fields and options the user can choose from to create constraints on the data to their liking.

Project Milestones

1. Generating data; implementing column constraints (UNIQUE, CHECK, ranges, selectivity)
    a. Premade columns
    b. Custom columns built on common SQL data types: VARCHAR, INT, DATE, etc
2. Building a table; implementing within-table constraints (CHECK)
3. Building a database; implementing between-table constraints (foreign keys, etc)
4. Developing front end for application using Python Tkinter

Characters

When it comes to generating realistic data, the character data type plays a crucial role in defining the attributes of the data. While the considerations between "CHAR" vs "VARCHAR" and length of the datatype are the only considerations when defining the data type in a table in SQL, generating realistic data requires us to consider the types of data that we want to generate. For example, when generating data for names, we need to ensure that the generated data is plausible and representative of real names. Similarly, when generating data for addresses and emails, we need to ensure that the generated data is valid and realistic. Additionally, participation constraints and join selectivity also need to be considered when generating realistic data to ensure that the generated data is consistent with the other data in the database and can be effectively joined with other tables. To facilitate characteristics that follow this requirement, we create names and emails together such that the email address contains the name of the corresponding user. We use the Faker python library to generate names, addresses and emails. To enhance the ability of the generated char data to be widely applicable, we have planned to incorporate another feature. This feature will allow the user to specify certain characteristics of the generated data such as the length, and whether the index contains a letter, or a digit. By giving users the ability to customize the generated char data, we hope to make it more flexible and adaptable to a wider range of use cases.

Integers

Uniform Distribution: users must input (i) number of values to generate, (ii) possible minimum value in range, (iii) possible max value in range, (iv) if min value must be present in dataset, (v) if maximum value must be present in dataset, (vi) selectivity percentage i.e. probability that any row is a particular value. The app will generate all integers within the range with equal probability. For example, to generate mobile phone numbers in Singapore (9[0-8]XXXXXX or 8XXXXXXX), users will indicate min value of 80000000, max value of 98999999 and other arguments as desired.

Floats

Uniform Distribution: users must input (i) number of values to generate, (ii) possible minimum value in range, (iii) possible maximum value in range, (iv) selectivity percentage i.e. probability that any row is a particular value, (v) number of digits after decimal point. For example, to generate prices, users will specify 2 digits after the decimal point. This generator works by populating with random samples from a uniform distribution over [0,1) range before transforming the values to fit over the user-defined range, then rounding them to the indicated number of decimal points.

Normal, Poisson Distributions: users input the relevant distribution parameters (mean and standard deviation for *Normal Distribution*; lambda mean for *Poisson Distribution*). However, the generators for these distributions are limited by the inability to specify minimum and maximum values as well as the selectivity percentage of the dataset.

Dates and Times

Uniform Distribution: users will firstly input the number of Dates, Times or Date-Times to generate. They will have the option to include the lower and upper bound of the value of the respective data-type they request for. The default bound for Dates will be set from 30 years from the current date to the current date. They will also have the option to include the mean value and standard deviation.

Normal Distribution: users will input the upper and lower bound of the date/time/date-time to generate. They will also have the option to include the mean value and standard deviation, so as to create a normal distribution, before generating a time using the properties of the normal distribution curve generated.