Generating Rich People Data and Corresponding Wealth Statistics in an SQLite Database

Student Name: Abhishek Aeera

Student Id: 22097107

GitHub: https://github.com/AbhishekAeera/Richest-people

Introduction:

In this project, we delve into the realm of data simulation and database management to construct a comprehensive dataset focusing on the world's wealthiest individuals and their corresponding wealth statistics. Leveraging Python and SQLite, we embark on the task of generating synthetic data to populate two distinct tables within our database: "Richest People" and "Wealth Statistics". The "Richest People" table encapsulates a plethora of information about affluent individuals, encompassing attributes such as their names, ages, net worth, sources of wealth, citizenships, and industries they belong to. On the other hand, the "Wealth Statistics" table aggregates pertinent data related to different wealth categories, providing insights into the average worth, total population, highest and lowest worth, dominant nationalities, and most common industries associated with these categories. By simulating this dataset, we lay the foundation for an analytical exploration into the intricate dynamics of wealth distribution and wealth creation across various demographic and economic dimensions. Through this project, we not only showcase proficiency in data generation and database management but also demonstrate the potential of synthesized datasets for insightful analysis and decision-making.

Data Generation.

The primary objective of this data generation process is to create a simulated dataset that mimics real-world data pertaining to wealthy individuals and wealth statistics. This dataset will serve as a foundation for analytical exploration and decision-making processes.

```
try:
    # Connect to SQLite database
    conn = sqlite3.connect('richest11_people.db')
    c = conn.cursor()
```

We generated synthetic data for wealthy individuals using Python's random module. Attributes such as name, age, net worth, source of wealth, citizenship, and industry were simulated.

Random names were generated by combining prefixes and suffixes. Age was generated using random integers within a specified range. Net worth was simulated as a random float within a predefined range, representing billions of dollars. Source of wealth, citizenship, and industry were randomly chosen from predefined lists.

```
import sqlite3
import random
import string

# Function to generate random names
def get_random_name():
    prefixes = ['John', 'Jane', 'Bob', 'Alice', 'Chris', 'Emma']
    suffixes = ['Smith', 'Johnson', 'Williams', 'Jones', 'Brown', 'Davis']
    return f"{random.choice(prefixes)} {random.choice(suffixes)}"

# Function to generate random richest people data
def generate_richest_people_data():
    name = get_random_name()
    age = random.randint(25, 90)
    net_worth = round(random.uniform(1, 200), 2)  # In billion dollars
    source_of_wealth = random.choice(["technology", "finance", "retail"])
    citizenship = random.choice(["USA", "China", "India", "Germany", "Brazil"])
    industry = random.choice(["IT", "Finance", "Retail"])
    return (name, age, net_worth, source_of_wealth, citizenship, industry)
```

Data for wealth statistics was generated to provide insights into various wealth categories. Wealth categories were randomly generated using a combination of uppercase letters. Wealth categories were randomly generated using a combination of uppercase letters. Data for each category was generated uniquely to avoid duplicates.

```
def generate_wealth_statistics_data(existing_categories):
    wealth_category = ''.join(random.choices(string.ascii_uppercase, k=5))
    while wealth_category in existing_categories:
        wealth_category = ''.join(random.choices(string.ascii_uppercase, k=5))
    existing_categories.add(wealth_category)
    average_worth = round(random.uniform(1, 200), 2)  # In billion dollars
    total_people = random.randint(100, 1000)
    highest_worth = round(random.uniform(100, 1000), 2)  # In billion dollars
    lowest_worth = round(random.uniform(1, 100), 2)  # In billion dollars
    dominant_nationality = random.choice(["USA", "China", "India", "Germany", "Brazil"])
    most_common_industry = random.choice(["IT", "Finance", "Retail"])
    return (wealth_category, average_worth, total_people, highest_worth, lowest_worth, dominant_nationality, most_common_indu

# Set to store existing wealth categories
existing_categories = set()
```

Ensuring uniqueness: To prevent duplicate entries, particularly in the Wealth Statistics table, we implemented checks to ensure that each wealth category is unique. Balancing randomness: While randomness is essential for creating realistic data, ensuring a balance between randomness and coherence was crucial to generate meaningful insights. The data generation process successfully created a dataset containing 1000 records for both the Richest People and Wealth Statistics tables. The dataset provides a diverse representation of wealthy individuals and wealth categories, enabling comprehensive analysis and exploration. Incorporating more attributes: Future iterations of the data generation process could include additional attributes for richer analysis, such as geographic location, education level, or philanthropic activities. Refinement of randomness: Fine-tuning the randomness parameters could lead to more nuanced and realistic datasets. Integration with real-world data: Combining simulated data with real-world datasets could provide deeper insights and enhance the dataset's applicability in various domains.

The data generation process successfully achieved its objectives by creating a comprehensive and diverse dataset for wealthy individuals and wealth statistics. This dataset serves as a valuable resource for analytical exploration and decision-making across various domains, laying the groundwork for further research and analysis in the field of wealth distribution and economics.

Database Schema

The database schema outlines the structure of the database, including tables, their attributes, and relationships between them. Here's the schema for the "Richest People" database:

Richest people Data Table:

```
# Create table for richest people data
c.execute('''CREATE TABLE IF NOT EXISTS RichestPeople (
                id INTEGER PRIMARY KEY,
                name TEXT,
                age INTEGER,
                net_worth REAL,
                source_of_wealth TEXT,
                citizenship TEXT,
                industry TEXT
            )''')
```

- id: INTEGER (Primary Key)
- name: TEXT
- age: INTEGER

- net worth: REAL
- source_of_wealth: TEXT
- citizenship: TEXT
- industry: TEXT

Wealth Statistics Table:

```
# Create WealthStatistics table
c.execute('''CREATE TABLE IF NOT EXISTS WealthStatistics (
                id INTEGER PRIMARY KEY,
                wealth_category TEXT UNIQUE,
                average_worth REAL,
                total_people INTEGER,
                highest_worth REAL,
                lowest_worth REAL,
                dominant_nationality TEXT,
                most_common_industry TEXT
            )''')
```

- id: INTEGER (Primary Key)
- wealth_category: TEXT (Unique)
- average_worth: REAL
- total_people: INTEGER
- highest_worth: REAL
- lowest_worth: REAL
- dominant_nationality: TEXT
- most_common_industry: TEXT

The **RichestPeople** table stores information about wealthy individuals, including their name, age, net worth, source of wealth, citizenship, and industry. The **WealthStatistics** table contains statistics related to various wealth categories. Each category is uniquely identified by its wealth_category attribute and includes information such as average worth, total number of people, highest and lowest worth, dominant nationality, and most common industry. There are no explicit relationships defined between the tables in this schema. However, both tables provide complementary information about wealth distribution, and analytical queries can be performed to derive insights from the combined dataset.

Justification and Ethical Discussion

Separating tables allows for better management of sensitive information, such as personal details of individuals in the "RichestPeople" table. Ethical considerations dictate that this data must be securely stored and accessed only by authorized personnel to prevent privacy breaches or misuse. Maintaining separate tables promotes transparency in data handling practices. It enables clear identification of data sources and facilitates accountability in data management procedures. Ethical guidelines require organizations to maintain accurate records and ensure transparency in reporting practices. Separating data into distinct categories helps mitigate biases and promotes fairness in analysis and decision-making processes. By analyzing aggregated wealth statistics separately from individual profiles, organizations can identify disparities and address potential biases in their wealth distribution models. Ethical data practices emphasize the importance of informed consent when collecting and using personal data. By segregating data into separate tables based on their intended use, organizations can provide clearer explanations to individuals regarding how their data will be utilized and seek appropriate consent for each purpose. In summary, the justification for separate tables lies in their ability to organize data effectively, support scalability and optimize query performance. Ethical considerations focus on safeguarding data privacy, promoting transparency and accountability, mitigating biases, and ensuring informed consent, thereby upholding ethical standards in data management practices.

Example Queries:

Query 1:

SQL 1

```
1  SELECT name, age, net_worth, source_of_wealth, citizenship, industry
2  FROM RichestPeople
3  WHERE age > 60
4  ORDER BY net_worth DESC;
5
```

	name	age	net_worth	source_of_wealth	citizenship	industry
1	Alice Williams	67	199.42	retail	USA	Finance
2	John Brown	68	199.04	finance	China	IT
3	Emma Williams	65	198.44	retail	Germany	Finance
4	John Smith	77	198.05	technology	USA	Finance
5	Alice Davis	80	197.79	finance	Germany	IT
6	John Johnson	71	197.48	finance	USA	Finance
7	John Johnson	64	197.47	technology	China	IT
8	Alice Jones	88	197.07	finance	USA	Finance

Executing this query provides insights into the demographics of wealthy individuals who are above the age of 60. By filtering based on age and sorting by net worth, we can identify older individuals who have accumulated significant wealth. This information can be valuable for various purposes, such as market research, financial planning, or demographic analysis.

Furthermore, this query demonstrates the capability of SQL to handle different data types, including text (name, source_of_wealth, citizenship, industry), numerical (age, net worth), and sorting based on numeric values (net worth). Additionally, the query showcases the use of selection criteria (age > 60) and ordering (ORDER BY net worth DESC) to retrieve specific subsets of data and present it in a meaningful manner.

Query 2:

SQL 1

```
1  SELECT wealth_category, AVG(average_worth) AS avg_worth, SUM(total_people) AS total_people
2  FROM WealthStatistics
3  GROUP BY wealth_category
4  ORDER BY avg_worth DESC;
5
6
```

	wealth_category	avg_worth	total_people
11	KYWSW	196.98	313
12	ZTFWR	196.46	202
13	JOKIC	195.96	471
14	CQVSN	195.81	618
15	VJPIP	195.59	745
16	TOZQO	195.54	143
17	QOWWM	195.45	246
18	DXKMC	195.32	110

Executing this SQL query provides insights into the average net worth and total number of people belonging to different wealth categories. By grouping the data by wealth category and calculating the average net worth and sum of total people within each category, we can analyse the distribution of wealth across various segments of the population. The results are ordered in descending order based on the average net worth within each wealth category. This allows us to identify the categories with

the highest average net worth, providing valuable information for understanding wealth distribution patterns. This query demonstrates the power of SQL in performing aggregate functions such as AVG() and SUM(), along with grouping data using GROUP BY. It enables us to summarize and analyze large datasets efficiently, facilitating decision-making processes in various fields such as economics, sociology, and finance.

Query 3:

SQL 1

```
1  SELECT rp.name, ws.dominant_nationality, rp.net_worth, ws.most_common_industry
2  FROM RichestPeople rp
3  JOIN WealthStatistics ws ON rp.citizenship = ws.dominant_nationality
4  WHERE rp.net_worth > 100 AND ws.total_people > 500;
5
6
7
```

	name	dominant_nationality	net_worth	most_common_industry
1	Chris Davis	India	159.05	Finance
2	Chris Davis	India	159.05	Retail
3	Chris Davis	India	159.05	Retail
4	Chris Davis	India	159.05	Finance
5	Chris Davis	India	159.05	Retail
6	Chris Davis	India	159.05	Retail
7	Chris Davis	India	159.05	Retail
8	Chris Davis	India	159.05	Finance

Executing this SQL query provides a comprehensive overview of wealthy individuals who share a common nationality with the dominant nationality in their respective wealth category. The query

retrieves data from both the richest People and wealth Statistics tables, joining them based on the citizenship of the individuals and the dominant nationality in the wealth statistics.The results include the name of the wealthy individual, their nationality, net worth, and the most common industry associated with their wealth category. By filtering the results to include only individuals with a net worth greater than $100 billion and wealth categories with more than 500 people, we focus on significant individuals and statistically relevant wealth categories.This query demonstrates the use of SQL joins to combine data from multiple tables based on a common attribute, allowing for more complex and insightful analysis. It provides valuable information for understanding the relationship between nationality, industry, and wealth distribution, contributing to broader discussions on socioeconomic factors and global wealth disparities.

Query 4:

SQL 1

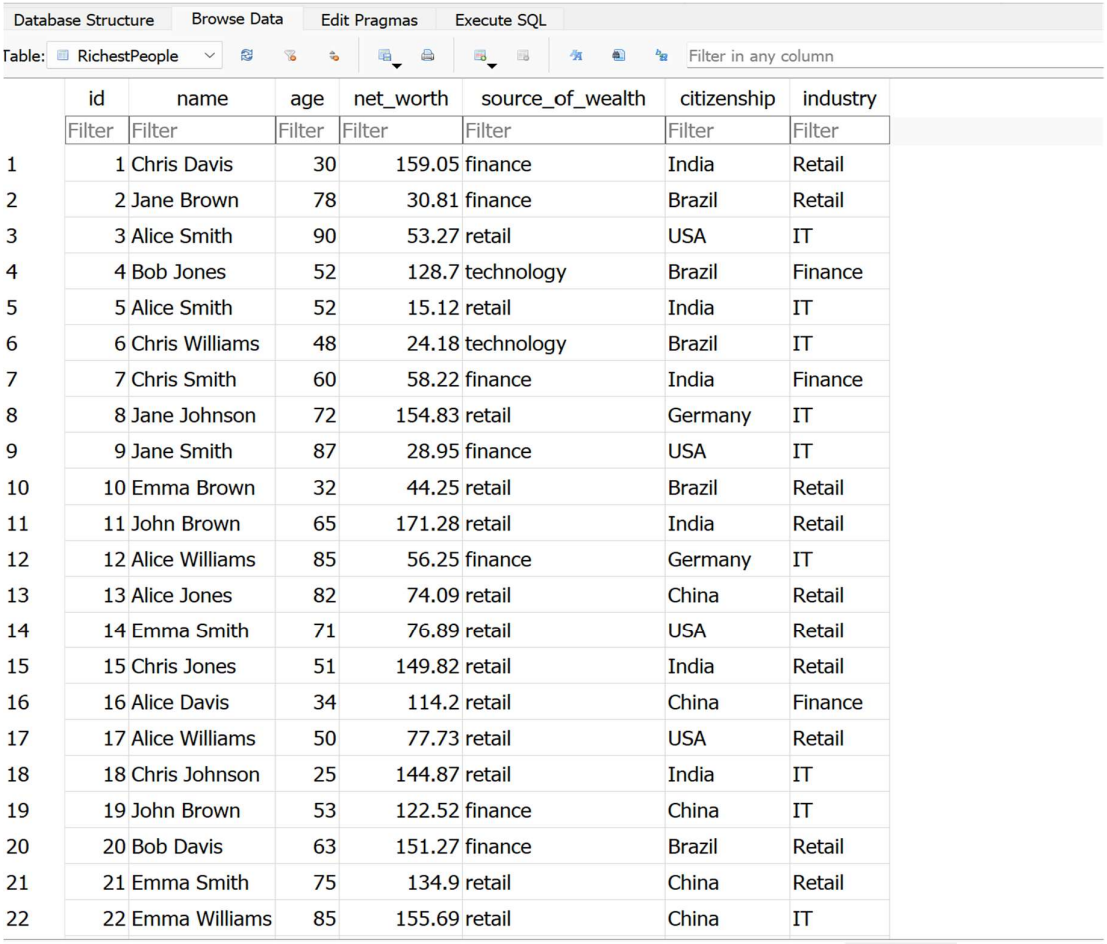
```
1  SELECT name, age, net_worth
2  FROM RichestPeople
3  WHERE net_worth > (
4      SELECT AVG(net_worth)
5      FROM RichestPeople
6  );
7
8
9
```

	name	age	net_worth
1	Chris Davis	30	159.05
2	Bob Jones	52	128.7
3	Jane Johnson	72	154.83
4	John Brown	65	171.28
5	Chris Jones	51	149.82
6	Alice Davis	34	114.2
7	Chris Johnson	25	144.87
8	John Brown	53	122.52

The SQL query selects the names, ages, and net worth of individuals from the richest People table whose net worth exceeds the average net worth of all individuals in the same table. Utilizing a

subquery within the WHERE clause, the average net worth is calculated dynamically, allowing for a comparison against individual net worth values. This query enables the identification of individuals whose wealth surpasses the average within the dataset, potentially highlighting outliers or exceptionally affluent individuals. Such analysis could be valuable for identifying trends, outliers, or conducting further investigation into the factors contributing to extreme wealth accumulation. By leveraging SQL's subquery capabilities, this query demonstrates the versatility of SQL in performing complex data analysis tasks within relational databases. It provides insights into the distribution of wealth within the dataset and aids in understanding the characteristics of individuals with significant net worth.

Conclusion:



Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: RichestPeople — Filter in any column

	id	name	age	net_worth	source_of_wealth	citizenship	industry
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Chris Davis	30	159.05	finance	India	Retail
2	2	Jane Brown	78	30.81	finance	Brazil	Retail
3	3	Alice Smith	90	53.27	retail	USA	IT
4	4	Bob Jones	52	128.7	technology	Brazil	Finance
5	5	Alice Smith	52	15.12	retail	India	IT
6	6	Chris Williams	48	24.18	technology	Brazil	IT
7	7	Chris Smith	60	58.22	finance	India	Finance
8	8	Jane Johnson	72	154.83	retail	Germany	IT
9	9	Jane Smith	87	28.95	finance	USA	IT
10	10	Emma Brown	32	44.25	retail	Brazil	Retail
11	11	John Brown	65	171.28	retail	India	Retail
12	12	Alice Williams	85	56.25	finance	Germany	IT
13	13	Alice Jones	82	74.09	retail	China	Retail
14	14	Emma Smith	71	76.89	retail	USA	Retail
15	15	Chris Jones	51	149.82	retail	India	Retail
16	16	Alice Davis	34	114.2	retail	China	Finance
17	17	Alice Williams	50	77.73	retail	USA	Retail
18	18	Chris Johnson	25	144.87	retail	India	IT
19	19	John Brown	53	122.52	finance	China	IT
20	20	Bob Davis	63	151.27	finance	Brazil	Retail
21	21	Emma Smith	75	134.9	retail	China	Retail
22	22	Emma Williams	85	155.69	retail	China	IT

Conclusion:

This table stores information about 1000 individuals, including their name, age, net worth, source of wealth, citizenship, and industry. The data is randomly generated to simulate a diverse population of wealthy individuals.

This table contains statistics related to wealth distribution, with each row representing a different wealth category. The statistics include the average net worth, total number of people, highest net worth, lowest net worth, dominant nationality, and most common industry within each wealth category. This data is also randomly generated.

The generated database can be used for various purposes such as statistical analysis, trend identification, and research on wealth distribution and its associated factors. By simulating diverse data, it provides a foundation for exploring patterns and correlations within the wealth landscape.

Overall, the code demonstrates the process of generating synthetic data and populating a SQLite database, which can serve as a valuable resource for further analysis and investigation in the realm of wealth distribution and related socio-economic phenomena.