

Snowflake – Test Data Generator

```
02-database-generator > 📁 generate-order.py > ...
1  # This script generates fake order data and inserts it into an SQLite database named 'retail_public.db'.
2  # It creates a table named 'Orders' if it doesn't already exist, generates 100 fake orders, and inserts them into
3  # The script also includes error handling for database operations and prints out the generated orders for verification.
4  from faker import Faker
5  import random
6  import sqlite3
7
8  # Initialize Faker
9  fake = Faker()
10
11 # Example customer IDs available in your database
12 customer_ids = [1, 2] # Replace with actual customer IDs from your Customers table
13
14 def generate_orders(num_orders):
15     """Function to generate fake order data."""
16     orders_data = []
17     for _ in range(num_orders):
18         customer_id = random.choice(customer_ids)
19         order_date = fake.date_between(start_date='-2y', end_date='today').strftime('%Y-%m-%d')
20         total_amount = round(random.uniform(20.0, 5000.0), 2) # Random amount between $20 and $5000
21         order = (customer_id, order_date, total_amount)
22         orders_data.append(order)
23     return orders_data
24
25 try:
26     # Connect to SQLite database
27     print("Connecting to the database...")
28     connection = sqlite3.connect('retail_public.db') # Ensure this path is correct
29     cursor = connection.cursor()
30
31     # Create table if it doesn't exist
32     print("Checking/creating the Orders table...")
33     cursor.execute(''
```

Tested – Data load – it is customizable.

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar:
 - SNOWFLAKE-CHATGPT** folder
 - .images
 - 01-vscode-assistant
 - 02-database-generator
 - 1-create-scripts.sql
 - 2-generate-customers.py
 - 3-insert-customers.sql
 - 4-insert-products.sql
 - 5-web-scraping.py
 - 6-web-scraping.csv
 - 7-select-queries.sql
 - generate-order.py
 - retail_public.db
 - RETAILDB.public
 - 03-metadata-inspector
 - app1.py
 - app2.py
 - app4.py
 - secrets.toml
 - OUTLINE
 - TIMELINE- 2-generate-customers.py** file content (selected in Explorer):

```
import random
fake = Faker()
customers = []
for i in range(100):
    customers.append({
        "CustomerKey": i,
        "FirstName": fake.first_name(),
        "LastName": fake.last_name(),
        "Email": fake.safe_email(),
        "Phone": fake.phone_number(),
        "AddressLine1": fake.street_address(),
        "AddressLine2": fake.secondary_address(),
        "City": fake.city(),
```
- TERMINAL** tab:

```
(venv) PS C:\DatamaticGit\SnowflakeChatGPT\snowflake-chatgpt\01-vscode-assistant> streamlit run .\leetcode-problem-solve-extended.py
here: https://aws.amazon.com/blogs/developer/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
Stopping...
● (venv) PS C:\DatamaticGit\SnowflakeChatGPT\snowflake-chatgpt\01-vscode-assistant> cd ..
● (venv) PS C:\DatamaticGit\SnowflakeChatGPT\snowflake-chatgpt> cd .\02-database-generator\
● (venv) PS C:\DatamaticGit\SnowflakeChatGPT\snowflake-chatgpt\02-database-generator> python .\generate-order.py
Connecting to the database...
Checking/creating the Orders table...
Generating sample orders...
Inserting orders into the database...
Records committed successfully.
Verifying records in the database...
(1, 2, '2025-09-19', 2739.26)
(2, 2, '2024-05-06', 4357.88)
(3, 1, '2024-11-27', 2020.45)
(4, 1, '2024-11-08', 908.88)
(5, 1, '2024-05-18', 1159.09)
```