Step 1: Add secret key Add the STRIPE_TEST_KEY into the Colab Secrets. Use the Secret key from your Stripe > Developers > API Keys section.

Step 2: Install required libraries Tap the run button to install the required python libraries

```
!pip install stripe==11.5.0
```

```
Collecting stripe==11.5.0
  Downloading stripe-11.5.0-py2.py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: requests>=2.20 in
/usr/local/lib/python3.11/dist-packages (from stripe==11.5.0) (2.32.3)
Requirement already satisfied: typing-extensions>=4.5.0 in
/usr/local/lib/python3.11/dist-packages (from stripe==11.5.0) (4.13.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.20-
>stripe==11.5.0) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.20-
>stripe==11.5.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.20-
>stripe==11.5.0) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.20-
>stripe==11.5.0) (2025.4.26)
Downloading stripe-11.5.0-py2.py3-none-any.whl (1.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 14.2 MB/s eta
0:00:00
pting uninstall: stripe
    Found existing installation: stripe 12.1.0
    Uninstalling stripe-12.1.0:
      Successfully uninstalled stripe-12.1.0
Successfully installed stripe-11.5.0
```

```
{"id":"e61e0168a908490ca767ef2f81745f92","pip_warning":{"packages":
["stripe"]}}
```

Step 3: Run the script to create the test data Tap run to insert data.

```python
import stripe
import random
from google.colab import userdata

SECRET_KEY = userdata.get("STRIPE_TEST_KEY")

stripe.api_key = SECRET_KEY
```

```python
# Sample data for generating random names
first_names = ("Alice", "Bob", "Charlie", "Diana", "Eve", "Frank",
"Grace", "Hank", "Ivy", "Jack", "Quinton", "Akriti", "Justin",
"Marcos")
last_names = ("Smith", "Johnson", "Williams", "Jones", "Brown",
"Davis", "Miller", "Wilson", "Moore", "Taylor", "Wall", "Chau",
"Keswani", "Marx")
# Sample clothing product names
clothing_names = (
    "T-Shirt", "Jeans", "Jacket", "Sweater", "Hoodie",
    "Shorts", "Dress", "Blouse", "Skirt", "Pants",
    "Shoes", "Sandals", "Sneakers", "Socks", "Hat",
    "Scarf", "Gloves", "Coat", "Belt", "Tie",
    "Tank Top", "Cardigan", "Overalls", "Tracksuit", "Polo Shirt",
    "Cargo Pants", "Capris", "Dungarees", "Boots", "Cufflinks",
    "Raincoat", "Peacoat", "Blazer", "Slippers", "Underwear",
    "Leggings", "Windbreaker", "Tracksuit Bottoms", "Beanie", "Bikini"
)
# List of random colors
colors = (
    "Red", "Blue", "Green", "Yellow", "Black", "White", "Gray",
    "Pink", "Purple", "Orange", "Brown", "Teal", "Navy", "Maroon",
    "Gold", "Silver", "Beige", "Lavender", "Turquoise", "Coral"
)

def create_customers(count=5):
    customers = []
    for _ in range(count):
        first_name = random.choice(first_names)
        last_name = random.choice(last_names)
        name = f"{first_name} {last_name}"
        email = f"{first_name.lower()}.
{last_name.lower()}@example.com"

        customer = stripe.Customer.create(
            name=name,
            email=email,
            description="Sample customer for testing"
        )
        customers.append(customer)
        print(f"Created Customer: {customer['name']} (ID:
{customer['id']})")
    return customers


def create_products(count=3):
    products = []
    for _ in range(count):
        color = random.choice(colors)
        product_name = random.choice(clothing_names)
        full_name = f"{color} {product_name}"
        product = stripe.Product.create(
            name=full_name,
            description=f"This is a {color.lower()}
{product_name.lower()}"
        )
        products.append(product)
        print(f"Created Product: {product['name']} (ID:
{product['id']})")
    return products


def create_prices(products, min_price=500, max_price=5000):
    prices = []
    for product in products:
        unit_amount = random.randint(min_price, max_price)  # Random
amount in cents
        price = stripe.Price.create(
            unit_amount=unit_amount,
            currency="usd",
            product=product['id']
        )
        prices.append(price)
        print(f"Created Price: ${unit_amount / 100:.2f} for Product
{product['name']} (ID: {price['id']})")
    return prices


def create_invoices(customers, prices, max_invoices_per_customer=5):
    invoices = []
    for customer in customers:
        num_invoices = random.randint(1, max_invoices_per_customer)
        for _ in range(num_invoices):
            price = random.choice(prices)
            # Create an invoice item first
            invoice_item = stripe.InvoiceItem.create(
                customer=customer['id'],
                price=price['id'],
                description=f"Purchase of {price['product']}"
            )

            # Create the invoice
            invoice = stripe.Invoice.create(
                customer=customer['id'],
                auto_advance=False,  # Don't auto-finalize
                collection_method='charge_automatically',
                description=f"Invoice for {customer['name']}"
            )

            # Finalize and pay the invoice manually
            invoice = stripe.Invoice.finalize_invoice(invoice['id'])
            if invoice.status != 'paid':  # Only pay if not already
paid
                invoice = stripe.Invoice.pay(invoice['id'])

            invoices.append(invoice)
            print(f"Created Invoice for Customer {customer['name']}
(Amount: ${price['unit_amount'] / 100:.2f})")
    return invoices

def main():
    print("Creating sample customers with random names...")
    customers = create_customers(count=10)
    print("Creating sample products with random clothing names and
colors...")
    products = create_products(count=15)
    print("Creating prices for products with random amounts...")
    prices = create_prices(products, min_price=500, max_price=5000)
    print("Creating random invoices for each customer...")
    invoices = create_invoices(customers, prices,
max_invoices_per_customer=5)
    print("Sample data creation complete!")
    print(f"Created {len(customers)} customers, {len(products)}
products, and {len(invoices)} invoices.")

main()
```

```
Creating sample customers with random names...
Created Customer: Justin Moore (ID: cus_SN4FWm0h6ke041)
Created Customer: Hank Marx (ID: cus_SN4FpxIlFGD0dT)
Created Customer: Ivy Johnson (ID: cus_SN4FVfR7gGD2FV)
Created Customer: Eve Marx (ID: cus_SN4FwbEoJE94Cc)
Created Customer: Hank Miller (ID: cus_SN4FeO7ka8JVFp)
Created Customer: Frank Smith (ID: cus_SN4F8ZzHW4N1mr)
Created Customer: Ivy Chau (ID: cus_SN4FiJVVFfXrkH)
Created Customer: Hank Chau (ID: cus_SN4FCafT0GAnEF)
Created Customer: Alice Marx (ID: cus_SN4FnNUc4D9k30)
Created Customer: Frank Miller (ID: cus_SN4FcPfiPmoeYM)
Creating sample products with random clothing names and colors...
Created Product: Yellow Windbreaker (ID: prod_SN4FzeWTK2ZeJp)
Created Product: White Belt (ID: prod_SN4FA7gU0YBWpu)
Created Product: Blue Cardigan (ID: prod_SN4FN3OkCXhiaq)
Created Product: Red Hat (ID: prod_SN4Fey3Vrer0Og)
Created Product: Pink Tank Top (ID: prod_SN4F2TZh9wuTDb)
Created Product: Red Socks (ID: prod_SN4F8k90X5XIYp)
Created Product: Green Boots (ID: prod_SN4FTHyurwZlex)
Created Product: Turquoise Leggings (ID: prod_SN4FlqeWJVQEPQ)
Created Product: Gray Belt (ID: prod_SN4FXAWjrmc1rr)
Created Product: Gold Slippers (ID: prod_SN4FAtN4AgBFTa)
Created Product: White Pants (ID: prod_SN4FovTZl4uc8P)
Created Product: Silver Boots (ID: prod_SN4FmXVQ81J2Ef)
Created Product: Purple Leggings (ID: prod_SN4Fc0nCjagUOe)
Created Product: Green Sweater (ID: prod_SN4FhZXOoegYa7)
Created Product: Yellow Tank Top (ID: prod_SN4F7yPrFPR0CL)
Creating prices for products with random amounts...
Created Price: $22.44 for Product Yellow Windbreaker (ID:
price_1RSK6sCHi0mHgAq3JI85nYZV)
Created Price: $23.02 for Product White Belt (ID:
price_1RSK6sCHi0mHgAq3F139OfJF)
Created Price: $10.72 for Product Blue Cardigan (ID:
price_1RSK6sCHi0mHgAq36Tb1SU9g)
Created Price: $9.38 for Product Red Hat (ID:
price_1RSK6sCHi0mHgAq3phtGaP5E)
Created Price: $36.43 for Product Pink Tank Top (ID:
price_1RSK6tCHi0mHgAq39s31LgWA)
Created Price: $24.27 for Product Red Socks (ID:
price_1RSK6tCHi0mHgAq3rp8q8eZj)
Created Price: $20.02 for Product Green Boots (ID:
price_1RSK6tCHi0mHgAq3hDgN4qKw)
Created Price: $29.00 for Product Turquoise Leggings (ID:
price_1RSK6tCHi0mHgAq3vI0PWhQD)
Created Price: $10.29 for Product Gray Belt (ID:
price_1RSK6tCHi0mHgAq3ud0y1qy4)
Created Price: $34.63 for Product Gold Slippers (ID:
price_1RSK6tCHi0mHgAq3D1TEbmFH)
Created Price: $26.12 for Product White Pants (ID:
price_1RSK6uCHi0mHgAq3n93KBTAP)
Created Price: $17.79 for Product Silver Boots (ID:
price_1RSK6uCHi0mHgAq3t4wYhZ1v)
Created Price: $5.68 for Product Purple Leggings (ID:
price_1RSK6uCHi0mHgAq3FnXAzKoN)
Created Price: $48.10 for Product Green Sweater (ID:
price_1RSK6uCHi0mHgAq3TjEWQPsd)
Created Price: $41.48 for Product Yellow Tank Top (ID:
price_1RSK6uCHi0mHgAq39r52dtkf)
Creating random invoices for each customer...
Created Invoice for Customer Justin Moore (Amount: $24.27)
Created Invoice for Customer Justin Moore (Amount: $41.48)
Created Invoice for Customer Justin Moore (Amount: $17.79)
Created Invoice for Customer Hank Marx (Amount: $10.29)
Created Invoice for Customer Ivy Johnson (Amount: $29.00)
Created Invoice for Customer Ivy Johnson (Amount: $48.10)
Created Invoice for Customer Ivy Johnson (Amount: $41.48)
Created Invoice for Customer Ivy Johnson (Amount: $20.02)
Created Invoice for Customer Eve Marx (Amount: $34.63)
Created Invoice for Customer Eve Marx (Amount: $10.29)
Created Invoice for Customer Eve Marx (Amount: $23.02)
Created Invoice for Customer Hank Miller (Amount: $29.00)
Created Invoice for Customer Hank Miller (Amount: $10.29)
Created Invoice for Customer Frank Smith (Amount: $36.43)
Created Invoice for Customer Frank Smith (Amount: $5.68)
Created Invoice for Customer Frank Smith (Amount: $41.48)
Created Invoice for Customer Frank Smith (Amount: $22.44)
Created Invoice for Customer Ivy Chau (Amount: $24.27)
Created Invoice for Customer Hank Chau (Amount: $23.02)
Created Invoice for Customer Hank Chau (Amount: $9.38)
Created Invoice for Customer Alice Marx (Amount: $41.48)
Created Invoice for Customer Alice Marx (Amount: $17.79)
Created Invoice for Customer Alice Marx (Amount: $20.02)
Created Invoice for Customer Frank Miller (Amount: $10.29)
Created Invoice for Customer Frank Miller (Amount: $48.10)
Created Invoice for Customer Frank Miller (Amount: $10.29)
Sample data creation complete!
Created 10 customers, 15 products, and 26 invoices.
```