# **Project Guide: Real-Time Credit Breach Dashboard with Snowflake + Streamlit**

## **Project Overview**

### **🌐 Goal:** Build a live, interactive dashboard to monitor credit breaches (like PayPal), using:

* 🏦 **Snowflake Unistore (Hybrid Tables)** — stores transactions + alerts
* 🎨 **Streamlit** — simple Python-based UI framework

## **🧰 Prerequisites**

|  |  |  |
| --- | --- | --- |
| **Tool** | **Why Needed** | **Install** |
| ✅ Snowflake Trial Account | Your cloud data warehouse | [Sign up here](https://signup.snowflake.com/) |
| ✅ Python 3.8+ | To run Streamlit | [Download Python](https://www.python.org/downloads/) |
| ✅ Streamlit | For UI/dashboard | pip install streamlit |
| ✅ Snowflake Python Connector | To run SQL from Python | pip install snowflake-connector-python |
| ✅ SnowSQL (optional) | CLI access to Snowflake | [Install](https://docs.snowflake.com/en/user-guide/snowsql-install-config) |

## **🗂️ Project Folder Structure**

snowflake-streamlit-credit-alert/

│

├── .env # Snowflake credentials

├── app.py # Main Streamlit app

├── snowflake\_init.sql # Snowflake schema + data script

├── requirements.txt # Python packages

└── README.md # Project instructions

## **🥇 STEP 1: Create a Free Snowflake Account**

1. Go to:<https://signup.snowflake.com/>
2. Choose **Standard Edition**, **AWS** or **GCP**, and region nearest to you.
3. After signup, you’ll receive:
   * Your **account name** → like abcde-xy123
   * Your **login URL**
   * Create a **username & password**

## **🧾 STEP 2: Get Your Snowflake Connection Info**

You'll need the following:

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Where to Find** |
| user | Your Snowflake username | You set this |
| password | Your password | You set this |
| account | Your account locator (e.g. xy12345.ap-southeast-1) | From login URL |
| warehouse | Compute warehouse name | Create manually if needed |
| database | Logical DB for your project | We'll create it |
| schema | Table grouping | We'll use public |

## 

## 

## **🛠️ STEP 3: Set Up Environment Locally**

### **A. Clone/Download Project Folder**

git clone https://github.com/your-repo/snowflake-streamlit-credit-alert.git

cd snowflake-streamlit-credit-alert

OR just create files manually in a folder.

### **B. Create .env File (store Snowflake credentials)**

.env:

SNOWFLAKE\_USER=your\_user

SNOWFLAKE\_PASSWORD=your\_password

SNOWFLAKE\_ACCOUNT=your\_account\_id

SNOWFLAKE\_WAREHOUSE=COMPUTE\_WH

SNOWFLAKE\_DATABASE=PAYPAL\_DB

SNOWFLAKE\_SCHEMA=PUBLIC

❗ Don’t commit this to GitHub! It contains secrets.

### **C. Install Required Python Packages**

pip install -r requirements.txt

**requirements.txt:**

streamlit

pandas

snowflake-connector-python

python-dotenv

## 

## 

## 

## **🧱 STEP 4: Create Snowflake Tables and Load Test Data**

### **Run the SQL Init Script**

Open **Snowflake Web UI** (from login URL), and paste contents of snowflake\_init.sql:

**snowflake\_init.sql**:

-- Create DB and tables

CREATE OR REPLACE DATABASE PAYPAL\_DB;

USE PAYPAL\_DB;

CREATE OR REPLACE SCHEMA PUBLIC;

USE SCHEMA PUBLIC;

CREATE OR REPLACE TABLE customers (

customer\_id STRING PRIMARY KEY,

name STRING,

credit\_limit NUMBER

);

INSERT INTO customers VALUES

('CUST001', 'Alice', 1000),

('CUST002', 'Bob', 500),

('CUST003', 'Charlie', 750);

CREATE OR REPLACE HYBRID TABLE hybrid\_transactions (

txn\_id STRING PRIMARY KEY,

customer\_id STRING,

amount NUMBER,

txn\_time TIMESTAMP,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

INSERT INTO hybrid\_transactions VALUES

('TXN1001', 'CUST001', 300, CURRENT\_TIMESTAMP),

('TXN1002', 'CUST001', 400, CURRENT\_TIMESTAMP),

('TXN1003', 'CUST002', 600, CURRENT\_TIMESTAMP);

CREATE OR REPLACE HYBRID TABLE credit\_breach\_alerts (

alert\_id STRING PRIMARY KEY,

customer\_id STRING,

total\_spent NUMBER,

credit\_limit NUMBER,

breach\_time TIMESTAMP

);

CREATE OR REPLACE TABLE daily\_summary AS

SELECT

customer\_id,

DATE(txn\_time) AS txn\_date,

SUM(amount) AS daily\_spend,

COUNT(\*) AS txn\_count

FROM hybrid\_transactions

GROUP BY customer\_id, DATE(txn\_time);

## **🖼️ STEP 5: Run the Streamlit App Locally**

### **🐍 Create app.py — the Streamlit App**

import streamlit as st

import pandas as pd

import snowflake.connector

from dotenv import load\_dotenv

import os

load\_dotenv()

# Snowflake credentials

conn = snowflake.connector.connect(

user=os.getenv("SNOWFLAKE\_USER"),

password=os.getenv("SNOWFLAKE\_PASSWORD"),

account=os.getenv("SNOWFLAKE\_ACCOUNT"),

warehouse=os.getenv("SNOWFLAKE\_WAREHOUSE"),

database=os.getenv("SNOWFLAKE\_DATABASE"),

schema=os.getenv("SNOWFLAKE\_SCHEMA")

)

st.set\_page\_config(page\_title="Credit Breach Dashboard", layout="wide")

st.title("🚨 Real-Time Credit Breach Monitor - PayPal")

# Fetch alert data

cur = conn.cursor()

cur.execute("SELECT \* FROM credit\_breach\_alerts ORDER BY breach\_time DESC")

alerts = cur.fetchall()

df\_alerts = pd.DataFrame(alerts, columns=['alert\_id', 'customer\_id', 'total\_spent', 'credit\_limit', 'breach\_time'])

# Filters

if not df\_alerts.empty:

selected\_customer = st.selectbox("Select Customer", df\_alerts['customer\_id'].unique())

st.dataframe(df\_alerts[df\_alerts['customer\_id'] == selected\_customer])

else:

st.warning("No alerts yet.")

# Trendline chart

cur.execute("SELECT customer\_id, txn\_date, daily\_spend FROM daily\_summary")

summary = cur.fetchall()

df\_summary = pd.DataFrame(summary, columns=['customer\_id', 'txn\_date', 'daily\_spend'])

chart\_data = df\_summary[df\_summary['customer\_id'] == selected\_customer]

st.line\_chart(chart\_data.set\_index('txn\_date')['daily\_spend'])

# Export

st.download\_button("📥 Download All Alerts", df\_alerts.to\_csv(), "alerts.csv", "text/csv")

### 

### **▶️ Run the App:**

streamlit run app.py

✅ Your browser should open at http://localhost:8501  
 You’ll see the dashboard live, pulling real-time data from Snowflake.

## **☁️ (Optional) STEP 6: Deploy to Streamlit Cloud (Free)**

1. Push code to GitHub
2. Go to<https://streamlit.io/cloud>
3. Connect to GitHub
4. Choose your repo
5. Set environment variables (from .env)
6. Deploy

💡 Use [Streamlit Secrets](https://docs.streamlit.io/streamlit-cloud/secrets-management) for credentials in production.

## **📘 Final Notes**

* You now have a **fully working real-time FinTech monitoring dashboard**.
* The logic and UI run **entirely on Python + SQL**, no web dev needed.
* You’ve built:
  + ✅ A hybrid OLTP/OLAP system
  + ✅ A live dashboard powered by Snowflake
  + ✅ An alert system for business risk management

**Happy learning**

**Best Regards**

**Saransh Jain**