# **Guided Lab: Incremental ETL with Streams + Tasks: PayPal Use Case (end to end pipeline)**

## **🚀 Objective:**

Simulate a real-time user signup pipeline for a FinTech company (PayPal) where incremental signups are tracked, audited, and archived using Snowflake **Streams** and **Tasks**.

### **🌟 Key Concepts:**

· Streams: CDC (Change Data Capture) object to track new rows

· Tasks: Scheduled jobs to automate incremental loads

· Warehouses: Configure size and auto-resume/suspend for cost-efficient automation

## **🔍 Real-World Problem: Audit User Signups in Real-Time**

**Use Case:** PayPal needs to: - Capture user signups in near real-time - Store raw entries for compliance - Move validated records into an audit table using automation

Snowflake solves this with: - A staging table for raw ingest - A stream to track changes - A task to move changes into an audit table every minute

## **🔺 Step-by-Step Implementation**

### **✅ Step 1: Setup Environment**

USE ROLE SYSADMIN;  
 CREATE OR REPLACE DATABASE PAYPAL\_ANALYTICS;  
 CREATE OR REPLACE SCHEMA SIGNUP\_PIPELINE;  
 USE SCHEMA PAYPAL\_ANALYTICS.SIGNUP\_PIPELINE;

### 

### **✅ Step 2: Create Raw and Audit Tables**

-- Raw signup data from app/middleware  
 CREATE OR REPLACE TABLE user\_signup\_raw (  
 user\_id STRING,  
 full\_name STRING,  
 email STRING,  
 signup\_ts TIMESTAMP,  
 country STRING  
 );  
  
 -- Processed signup records for downstream analytics  
 CREATE OR REPLACE TABLE user\_signup\_audit (  
 user\_id STRING,  
 full\_name STRING,  
 email STRING,  
 signup\_ts TIMESTAMP,  
 country STRING,  
 processed\_ts TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
 );

### **✅ Step 3: Create a Stream**

-- Create a stream to capture inserts in raw signup table  
 CREATE OR REPLACE STREAM signup\_stream ON TABLE user\_signup\_raw;

#### **🔹 Explanation:**

The stream tracks **only new inserts** and exposes metadata like METADATA$ACTION, METADATA$ISUPDATE, etc.

### **✅ Step 4: Create Task to Automate Insert**

-- Task runs every 1 minute using XS warehouse  
 CREATE OR REPLACE TASK signup\_task  
 WAREHOUSE = 'XS\_WH'  
 SCHEDULE = '1 MINUTE'  
 AS  
 INSERT INTO user\_signup\_audit (user\_id, full\_name, email, signup\_ts, country)  
 SELECT user\_id, full\_name, email, signup\_ts, country  
 FROM signup\_stream  
 WHERE METADATA$ACTION = 'INSERT';

#### **🔹 Explanation:**

This task picks only new signups and inserts into the audit table. It runs every minute.

### 

### **✅ Step 5: Insert Sample Data to Simulate Traffic**

INSERT INTO user\_signup\_raw (user\_id, full\_name, email, signup\_ts, country)  
 VALUES  
 ('U001', 'Akhil Reddy', 'akhil.reddy@paypal.com', CURRENT\_TIMESTAMP, 'IN'),  
 ('U002', 'Keerthi Nair', 'keerthi.nair@paypal.com', CURRENT\_TIMESTAMP, 'IN');

-- Wait 60 seconds for the task to execute.

### **✅ Step 6: Check the status and then activate the Task(if not activated)**

SHOW TASKS LIKE 'SIGNUP\_TASK'; *--Wait for the output and then run the next cmd if needed*

ALTER TASK signup\_task RESUME;

### **✅ Step 7: Validate Audit Table**

-- Check if records moved from raw to audit  
 SELECT \* FROM user\_signup\_audit;

### **✅ Optional: View Task and Stream Status**

-- View stream delta rows  
 SELECT \* FROM signup\_stream;  
  
 -- Check recent task runs  
 SELECT \* FROM TABLE(INFORMATION\_SCHEMA.TASK\_HISTORY())  
 WHERE NAME = 'SIGNUP\_TASK';

## **🏛️ Warehouse Optimization Notes**

| Scenario | Warehouse Config |
| --- | --- |
| Low-frequency signups | XS, auto-suspend enabled |
| High-load periods | S or M size |
| Multi-tenant tracking | Use separate tasks/streams for each app |

## **🚫 Cleanup (Optional)**

DROP TASK IF EXISTS signup\_task;  
 DROP STREAM IF EXISTS signup\_stream;  
 DROP TABLE IF EXISTS user\_signup\_raw;  
 DROP TABLE IF EXISTS user\_signup\_audit;  
 DROP SCHEMA IF EXISTS SIGNUP\_PIPELINE CASCADE;  
 DROP DATABASE IF EXISTS PAYPAL\_ANALYTICS;

## **📖 Summary**

| Feature | Used for |
| --- | --- |
| Stream | Track new signups |
| Task | Automate audit inserts |
| XS Warehouse | Cost-efficient automation |
| Audit Table | Permanent record for analytics |

✅ With this setup, PayPal can audit every signup in near real-time with full traceability and scalability.

**Happy learning**

**Regards**

**Saransh Jain**