**Exercise: Streams + Tasks + Snowpipe**

## Objective:

Create an end-to-end pipeline that ingests customer feedback CSV files using Snowpipe, tracks new rows with a Stream, and processes them automatically into a clean reporting table using a Task.

## Step 1: Sample CSV File

**File Name:** customer\_feedback.csv

**Example Data:**

feedback\_id,customer\_name,rating,comments,feedback\_date  
1,John Doe,5,Great service,2025-08-20 10:12:00  
2,Jane Smith,4,Good experience,2025-08-20 10:15:00  
3,Emily Jones,3,Average service,2025-08-20 10:18:00

## Step 2: Create Raw Table

CREATE TABLE raw\_feedback (  
 feedback\_id INT,  
 customer\_name STRING,  
 rating INT,  
 comments STRING,  
 feedback\_date TIMESTAMP  
);

## Step 3: Create Stream on Raw Table

CREATE STREAM raw\_feedback\_stream  
ON TABLE raw\_feedback;

## Step 4: Create Processed Table

CREATE TABLE processed\_feedback (  
 feedback\_id INT,  
 customer\_name STRING,  
 rating INT,  
 comments STRING,  
 feedback\_date TIMESTAMP  
);

## Step 5: Create Internal Stage

CREATE STAGE feedback\_stage;

## Step 6: Upload CSV File to Stage

PUT command

## Step 7: Create Snowpipe

CREATE PIPE feedback\_pipe  
AUTO\_INGEST = TRUE  
AS  
COPY INTO raw\_feedback  
FROM @feedback\_stage  
FILE\_FORMAT = (TYPE = 'CSV' FIELD\_OPTIONALLY\_ENCLOSED\_BY='"' SKIP\_HEADER=1);

## Step 8: Create Task to Process Stream

CREATE TASK process\_feedback\_task  
WAREHOUSE = COMPUTE\_WH  
SCHEDULE = '1 MINUTE'  
AS  
INSERT INTO processed\_feedback  
SELECT  
 feedback\_id,  
 UPPER(customer\_name),  
 rating,  
 comments,  
 feedback\_date  
FROM raw\_feedback\_stream;

* Resume the task:

ALTER TASK process\_feedback\_task RESUME;

## Step 9: Test the Pipeline

1. Upload another CSV file with new feedback rows to feedback\_stage.
2. Snowpipe automatically loads it into raw\_feedback.
3. Stream tracks new rows.
4. Task automatically inserts them into processed\_feedback.

Check processed table:

SELECT \* FROM processed\_feedback;

## Step 10: Bonus Exercises

1. Filter out rows with rating < 3 in the Task SQL.
2. Add a new column feedback\_length counting characters in comments.
3. Schedule the Task to run every 5 minutes instead of 1.
4. Create a materialized view on processed\_feedback to show average rating per day.