USN NUMBER: 1RVU23CSE227
NAME: Kruthik S

Ex No: 8 Date: 29-10-2025	**Implementing Real-Time Data Processing in Snowflake Using Streams and Tasks**

Objective:

To design and implement a real-time data processing pipeline in **Snowflake** using **Streams** and **Tasks**, automating the cleaning and transformation of raw hospital patient records into a clean, standardized dataset ready for analysis.

Outcomes:

1. Understand the working of **Streams** in tracking DML changes (INSERT, UPDATE, DELETE).
2. Learn to automate data transformations using **Tasks** in Snowflake.
3. Implement an end-to-end data cleaning pipeline for hospital patient records.
4. Schedule tasks to run automatically at fixed intervals for near real-time updates.
5. Validate the accuracy and consistency of cleaned data.

Materials

- **Tools:** Snowflake Web UI / Snowsight, SQL Worksheet.
- **Database:** HOSPITAL_DB
- **Schema:** PATIENT_SCHEMA
- **Tables:** RAW_PATIENTS, CLEAN_PATIENTS
- **Artifacts:** Stream and Task SQL scripts for automation.

Architecture

The architecture consists of:

1. **Raw Table:** Stores inconsistent, messy patient data.
2. **Stream:** Tracks changes (new inserts) from the raw table.
3. **Task:** Cleans and transforms streamed data and loads it into the clean table.
4. **Scheduler:** Runs the cleaning task every 10 minutes for real-time updates.

USN NUMBER: 1RVU23CSE227
NAME: Kruthik S

Lab Procedure

Step 1: Database and Schema Creation

```
CREATE DATABASE HOSPITAL_DB;
CREATE SCHEMA PATIENT_SCHEMA;
```

Step 2: Table Creation

```
CREATE OR REPLACE TABLE RAW_PATIENTS (
  patient_id STRING,
  name STRING,
  age STRING,
  gender STRING,
  visit_date STRING,
  diagnosis_code STRING,
  bill_amount STRING
);

CREATE OR REPLACE TABLE CLEAN_PATIENTS (
  patient_id INT,
  name STRING,
  age INT,
  gender STRING,
  visit_date DATE,
  diagnosis_code STRING,
  bill_amount FLOAT,
  processed_at TIMESTAMP_NTZ
);
```

Step 3: Stream Creation

```
CREATE OR REPLACE STREAM PATIENT_STREAM ON TABLE RAW_PATIENTS;
```

USN NUMBER: 1RVU23CSE227
NAME: Kruthik S

Step 4: Task for Data Cleaning

```
CREATE OR REPLACE TASK CLEAN_PATIENT_DATA_TASK
  WAREHOUSE = COMPUTE_WH
  SCHEDULE = '10 MINUTE'
AS
INSERT INTO CLEAN_PATIENTS
SELECT
  TRY_TO_NUMBER(patient_id) AS patient_id,
  name,
  COALESCE(TRY_TO_NUMBER(age), 0) AS age,
  gender,
  COALESCE(
    TRY_TO_DATE(visit_date, 'YYYY-MM-DD'),
    TRY_TO_DATE(visit_date, 'DD-MM-YYYY'),
    TRY_TO_DATE(visit_date, 'MM/DD/YYYY'),
    CURRENT_DATE()
  ) AS visit_date,
  diagnosis_code,
  COALESCE(TRY_TO_NUMBER(REPLACE(bill_amount, ',', ")), 0) AS bill_amount,
  CURRENT_TIMESTAMP() AS processed_at
FROM PATIENT_STREAM;
```

Step 5: Insert Messy Data

```
INSERT INTO RAW_PATIENTS VALUES
(1, 'John Doe', '30', 'Male', '2025-10-21', 'D01', '5,000'),
(2, 'Jane Smith', 'Twenty-Five', 'Female', '21-10-2025', 'D02', '3,200'),
(3, 'Alex Brown', NULL, 'Male', '2025/10/22', 'D03', 'abc'),
(4, 'Mary Lee', '40', NULL, NULL, 'D04', NULL);
```

Issues Identified:

- Non-numeric age ("Twenty-Five")
- Invalid bill amounts ("abc")
- Missing gender, visit dates
- Mixed date formats

Resolution by Cleaning Task:

- Converts text to INT, default 0 if invalid
- Replaces commas in bill amount, converts to FLOAT
- Parses dates in multiple formats
- Fills missing or invalid dates with current date

Step 6: Manual Task Execution

```
EXECUTE TASK CLEAN_PATIENT_DATA_TASK;
```

Step 7: Verification of Cleaned Data

```
SELECT * FROM CLEAN_PATIENTS ORDER BY patient_id;
```

Validation Checks:

- All age and bill_amount fields are valid numbers.
- visit_date correctly formatted as DATE.
- No missing or null critical fields after transformation.

Step 8: Extension

```
CREATE OR REPLACE TASK FLAG_MISSING_DIAGNOSIS
  WAREHOUSE = COMPUTE_WH
  SCHEDULE = '15 MINUTE'
AS
INSERT INTO REVIEW_FLAGS
SELECT * FROM CLEAN_PATIENTS
WHERE diagnosis_code IS NULL;
```

GitHub Link: https://github.com/kruth-s/Data-Engg-Lab