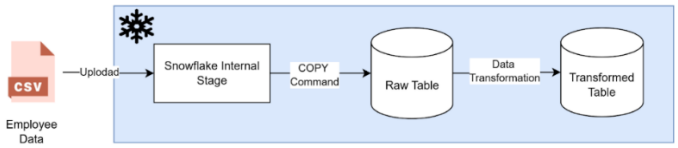
SNOWFLAKE - Mini Project 1



---

1. Create a Database & Schema
    1. Database : EMPLOYEE_DATA
        1. Schema : RAW_DATA
        2. Schema : TRANSFORMED_DATA

---

2. Create Raw Table in the RAW_DATA schema
    1. Table :
        1. Database: EMPLOYEE_DATA
        2. Schema: RAW_DATA
        3. Name: EMPLOYEE_RAW
    2. Table structure

        EMPLOYEE_ID STRING
        FIRST_NAME STRING
        LAST_NAME STRING
        DEPARTMENT STRING
        SALARY DECIMAL(10,2)
        HIRE_DATE DATE
        LOCATION STRING

---

3. Create Transformed Table in the TRANSFORMED_DATA schema
    1. Table :
        1. Database: EMPLOYEE_DATA
        2. Schema: TRANSFORMED_DATA
        3. Name: EMPLOYEE_TRANSFORMED
    2. Table structure

        EMPLOYEE_ID STRING
        FULL_NAME STRING
        DEPARTMENT STRING
        ANNUAL_SALARY DECIMAL(10, 2)
        HIRE_DATE DATE
        EXPERIENCE_LEVEL STRING
        TENURE_DAYS STRING
        STATE STRING
        COUNTRY STRING
        BONUS_ELIGIBILITY STRING
        HIGH_POTENTIAL_FLAG STRING

---

4. Upload the CSV File to an Internal Stage
    1. Create an Internal Stage in the RAW_DATA schema
        1. Database: EMPLOYEE_DATA
        2. Schema: RAW_DATA
        3. Name: EMPLOYEE_STAGE
    2. Upload the employee csv data file in the internal stage
    3. List the file in the internal stage

5. Load Data from the Stage into the Raw Table
    1. To load the raw employee data into the EMPLOYEE_RAW table, use the COPY INTO command

6. Perform Data Transformations and insert data into the Transformed Table
    1. Full Name: Concatenate first_name and last_name.
    2. Annual Salary: Multiply the monthly salary by 12.
    3. Experience Level: Classify employees based on the hire date. For example:
        1. New Hire: Less than 1 year.
        2. Mid-level: 1 to 5 years.
        3. Senior: More than 5 years.
    4. Employee Tenure: Calculate how long an employee has been with the company based on the hire_date in days
    5. State: Fetch the value before the hyphen(-) in the location column
    6. Country: Fetch the value after the hyphen(-) in the location column
    7. Employee's Eligibility for Bonus: For example, employees with a salary greater than $ 10,000 are eligible for a bonus.
    8. Flagging High-Potential Employees: Flag employees who have been with the company for more than 3 years.

7. Data Analysis on the transformed data
    1. Employee Count by Department
    2. Provide count of employees by country
    3. Extract employees who were hired within 12 months
    4. Extract the top 10% of employees by salary
    5. Calculate the total salary expense per department for each year.
    6. Determine how many employees with 5+ years with company

Solution:

```
-- SET ROLE AND WAREHOUSE:
USE ROLE ACCOUNTADMIN;
USE WAREHOUSE COMPUTE_WH;
```

```
-- 1) Create a Database & Schema:
CREATE OR REPLACE DATABASE EMPLOYEE_DATA;
USE DATABASE EMPLOYEE_DATA;
CREATE OR REPLACE SCHEMA RAW_DATA;
CREATE OR REPLACE SCHEMA TRANSFORMED_DATA;
```

```
-- 2) Create Raw Table in the RAW_DATA schema:
CREATE OR REPLACE TABLE RAW_DATA.EMPLOYEE_RAW (
EMPLOYEE_ID STRING,
FIRST_NAME STRING,
LAST_NAME STRING,
DEPARTMENT STRING,
SALARY DECIMAL(10,2),
HIRE_DATE DATE,
LOCATION STRING);
SELECT * FROM RAW_DATA.EMPLOYEE_RAW;
```

```sql
-- 3) Create Transformed Table in the TRANSFORMED_DATA schema:
CREATE OR REPLACE TABLE TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED (
EMPLOYEE_ID STRING,
FULL_NAME STRING,
DEPARTMENT STRING,
ANNUAL_SALARY DECIMAL(10, 2),
HIRE_DATE DATE,
EXPERIENCE_LEVEL STRING,
TENURE_DAYS STRING,
STATE STRING,
COUNTRY STRING,
BONUS_ELIGIBILITY STRING,
HIGH_POTENTIAL_FLAG STRING);
SELECT * FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED;
```

```sql
-- 4) Upload the CSV File to an Internal Stage:
-- Create an Internal Stage in the RAW_DATA schema:
CREATE OR REPLACE STAGE RAW_DATA.EMPLOYEE_STAGE;
LIST @EMPLOYEE_STAGE;

------- UPLOAD MANUALLY THE DATE INTO EMPLOYEE_STAGE -------
-------------------------------------------------------------
LIST @EMPLOYEE_STAGE;
```

```sql
-- 5) Load Data from the Stage into the Raw Table:
-- CREATE FILE_FORMAT FOR CSV:
CREATE OR REPLACE FILE FORMAT RAW_DATA.CSV_FORMAT
TYPE = 'CSV'
FIELD_DELIMITER = ','
RECORD_DELIMITER = '\n'
SKIP_HEADER = 1;

-- COPY STAGED DATA INTO TABLE:
SELECT * FROM RAW_DATA.EMPLOYEE_RAW;
COPY INTO RAW_DATA.EMPLOYEE_RAW
FROM @RAW_DATA.EMPLOYEE_STAGE
FILE_FORMAT = (FORMAT_NAME = CSV_FORMAT);
SELECT * FROM RAW_DATA.EMPLOYEE_RAW;
```

```sql
-- 6) Perform Data Transformations and insert data into the Transformed Table:
TRUNCATE TABLE TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED;
INSERT INTO TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED (
EMPLOYEE_ID, FULL_NAME, DEPARTMENT, ANNUAL_SALARY, HIRE_DATE, EXPERIENCE_LEVEL,
TENURE_DAYS, STATE, COUNTRY, BONUS_ELIGIBILITY, HIGH_POTENTIAL_FLAG)
SELECT
    EMPLOYEE_ID,

    -- 1) Full Name: Concatenate first_name and last_name.
    CONCAT(FIRST_NAME, ' ', LAST_NAME) AS FULL_NAME,
    DEPARTMENT,
```

```
    -- 2) Annual Salary: Multiply the monthly salary by 12.
    (12 * SALARY) AS ANNUAL_SALARY,
    HIRE_DATE,

    -- 3) Experience Level: Classify employees based on the hire date.
    CASE WHEN DATEDIFF(YEAR, HIRE_DATE, CURRENT_DATE) <= 1 THEN 'New Hire'
         WHEN DATEDIFF(YEAR, HIRE_DATE, CURRENT_DATE) <= 5 THEN 'Mid-level'
         ELSE 'Senior' END AS EXPERIENCE_LEVEL,
    -- 4) Employee Tenure: Calculate how long an employee has been with the company
based on the hire_date in days
    DATEDIFF(DAY, HIRE_DATE, CURRENT_DATE) AS TENURE_DAYS,
    -- 5) State: Fetch the value before the hyphen(-) in the location column
    SPLIT_PART(LOCATION, '-', 1) AS STATE,

    -- 6) Country: Fetch the value after the hyphen(-) in the location column
    SPLIT_PART(LOCATION, '-', -1) AS COUNTRY,

    -- 7) Employee's Eligibility for Bonus: For example, employees with a salary
greater than $ 10,000 are eligible for a bonus.
    CASE WHEN SALARY >= 10000 THEN 'YES' ELSE 'NO' END AS BONUS_ELIGIBILITY,

    -- 8) Flagging High-Potential Employees: Flag employees who have been with the
company for more than 3 years.
        CASE WHEN TENURE_DAYS / 365 >= 3.0 THEN 'YES' ELSE 'NO' END AS
HIGH_POTENTIAL_FLAG

FROM RAW_DATA.EMPLOYEE_RAW;
SELECT * FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED;
```

```
-- 6) Perform Data Transformations and insert data into the Transformed Table:
(Another method for cope tables)
TRUNCATE TABLE TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED;
INSERT INTO TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED
SELECT
    S.EMPLOYEE_ID,

    -- 1) Full Name: Concatenate first_name and last_name.
    CONCAT(S.FIRST_NAME, ' ', S.LAST_NAME) AS FULL_NAME,
    S.DEPARTMENT,

    -- 2) Annual Salary: Multiply the monthly salary by 12.
    (12 * S.SALARY) AS ANNUAL_SALARY,
    S.HIRE_DATE,

    -- 3) Experience Level: Classify employees based on the hire date.
    CASE WHEN DATEDIFF(YEAR, S.HIRE_DATE, CURRENT_DATE) <= 1 THEN 'New Hire'
         WHEN DATEDIFF(YEAR, S.HIRE_DATE, CURRENT_DATE) <= 5 THEN 'Mid-level'
         ELSE 'Senior' END AS EXPERIENCE_LEVEL,

    -- 4) Employee Tenure: Calculate how long an employee has been with the company
based on the hire_date in days
    DATEDIFF(DAY, S.HIRE_DATE, CURRENT_DATE) AS TENURE_DAYS,
```

```
    -- 5) State: Fetch the value before the hyphen(-) in the location column
    SPLIT_PART(LOCATION, '-', 1) AS STATE,

    -- 6) Country: Fetch the value after the hyphen(-) in the location column
    SPLIT_PART(LOCATION, '-', -1) AS COUNTRY,

    -- 7) Employee's Eligibility for Bonus: For example, employees with a salary
greater than $ 10,000 are eligible for a bonus.
    CASE WHEN S.SALARY >= 10000 THEN 'YES' ELSE 'NO' END AS BONUS_ELIGIBILITY,

    -- 8) Flagging High-Potential Employees: Flag employees who have been with the
company for more than 3 years.
        CASE WHEN TENURE_DAYS / 365 >= 3.0 THEN 'YES' ELSE 'NO' END AS
HIGH_POTENTIAL_FLAG

FROM RAW_DATA.EMPLOYEE_RAW S
LEFT JOIN TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED T
ON (S.EMPLOYEE_ID = T.EMPLOYEE_ID AND T.EMPLOYEE_ID IS NULL);
SELECT * FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED;
```

```
-- 7) Data Analysis on the transformed data

    -- 1) Employee Count by Department
    SELECT DEPARTMENT, COUNT(*) FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED GROUP BY
DEPARTMENT;

    -- 2) Provide count of employees by country
    SELECT COUNTRY, COUNT(*) FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED GROUP BY
COUNTRY ORDER BY COUNTRY;

    -- 3) Extract employees who were hired within 12 months
    SELECT EMPLOYEE_ID, FULL_NAME, HIRE_DATE FROM
TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED WHERE EXPERIENCE_LEVEL = 'New Hire' ORDER BY
HIRE_DATE;

    -- 4) Extract the top 10% of employees by salary
    SELECT EMPLOYEE_ID, FULL_NAME, ANNUAL_SALARY FROM
TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED ORDER BY ANNUAL_SALARY DESC LIMIT 10;

    -- 5) Calculate the total salary expense per department for each year.
    WITH CTE AS (
        SELECT  EMPLOYEE_ID, HIRE_DATE, ANNUAL_SALARY, DEPARTMENT,
                CASE WHEN YEAR(HIRE_DATE) > 2024 THEN 0 WHEN YEAR(HIRE_DATE) < 2024
THEN ANNUAL_SALARY ELSE -DATEDIFF(DAY, HIRE_DATE, '2024-01-01')*(ANNUAL_SALARY/365)
END AS SALAEY_2024,
                CASE WHEN YEAR(HIRE_DATE) > 2023 THEN 0 WHEN YEAR(HIRE_DATE) < 2023
THEN ANNUAL_SALARY ELSE -DATEDIFF(DAY, HIRE_DATE, '2023-01-01')*(ANNUAL_SALARY/365)
END AS SALAEY_2023,
                CASE WHEN YEAR(HIRE_DATE) > 2022 THEN 0 WHEN YEAR(HIRE_DATE) < 2022
THEN ANNUAL_SALARY ELSE -DATEDIFF(DAY, HIRE_DATE, '2022-01-01')*(ANNUAL_SALARY/365)
END AS SALAEY_2022,
```

```
                CASE WHEN YEAR(HIRE_DATE) > 2021 THEN 0 WHEN YEAR(HIRE_DATE) < 2021
THEN ANNUAL_SALARY ELSE -DATEDIFF(DAY, HIRE_DATE, '2021-01-01')*(ANNUAL_SALARY/365)
END AS SALAEY_2021,
                CASE WHEN YEAR(HIRE_DATE) > 2020 THEN 0 WHEN YEAR(HIRE_DATE) < 2020
THEN ANNUAL_SALARY ELSE -DATEDIFF(DAY, HIRE_DATE, '2020-01-01')*(ANNUAL_SALARY/365)
END AS SALAEY_2020,
                CASE WHEN YEAR(HIRE_DATE) > 2019 THEN 0 WHEN YEAR(HIRE_DATE) < 2019
THEN ANNUAL_SALARY ELSE -DATEDIFF(DAY, HIRE_DATE, '2019-01-01')*(ANNUAL_SALARY/365)
END AS SALAEY_2019,
                CASE WHEN YEAR(HIRE_DATE) > 2018 THEN 0 WHEN YEAR(HIRE_DATE) < 2018
THEN ANNUAL_SALARY ELSE -DATEDIFF(DAY, HIRE_DATE, '2018-01-01')*(ANNUAL_SALARY/365)
END AS SALAEY_2018
        FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED)
    SELECT DEPARTMENT, SUM(SALAEY_2024), SUM(SALAEY_2023), SUM(SALAEY_2022),
SUM(SALAEY_2021), SUM(SALAEY_2020), SUM(SALAEY_2019), SUM(SALAEY_2018)
    FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED JOIN CTE USING(EMPLOYEE_ID) GROUP BY
DEPARTMENT;

    -- 6) Determine how many employees with 5+ years with company
    SELECT COUNT(*) AS MORE_THAN_5_YEARS FROM TRANSFORMED_DATA.EMPLOYEE_TRANSFORMED
WHERE DATEDIFF(YEAR, HIRE_DATE, CURRE
```