
🔹 Task 1 – Employee Detail Load (Base Layer)

🎯 Objective

Extract and load detailed employee performance and compensation data from HR source tables into a target table (TGT_EMPLOYEE_DETAIL).

⚙️ Requirements

Source Tables:

- EMPLOYEES
- DEPARTMENTS
- JOBS
- LOCATIONS

Transformations to Implement:

1. **Source Qualifier (SQ_EMPLOYEES):**
    - Inner Join the four tables logically using the department, job, and location relationships.
    - Filter employees hired within the last 25 years. (25 years = 300 months)
    - Exclude any departments related to *Sales*.
2. **Expression Transformation (EXP_CALC):**
    - Create calculated columns:
        - Full Name => CONCAT first name + last name
        - Total Income (Salary + Commission) => salary + salary*commission

          If salary or commission null, please put 0
        - Years of Service (based on Hire Date) => ROUND(MONTHS_BETWEEN(SYSDATE, e.hire_date)/12, 2)
        - Salary Grade (High / Medium / Low)

          salary > 15000 THEN 'HIGH' - salary BETWEEN 8000 AND 15000 'MEDIUM' ELSE 'LOW'

3. **Filter Transformation (FIL_VALID):**
    - Exclude invalid or incomplete records (optional).
4. **Sorter Transformation (SRT_ORDER):**
    - Sort by Department ID (ascending) and Salary (descending).
5. **Target:**
    - Create and load data into a new target table named **TGT_EMPLOYEE_DETAIL.** (Target query definition below)
    - Expect around **72 records** after loading.

Deliverables:

- Mapping: m_EMPLOYEE_DETAIL_LOAD
- Session: s_EMPLOYEE_DETAIL_LOAD
- Workflow: wf_EMPLOYEE_DETAIL_LOAD
- Validation: Target row count = 72

🔹 Task 2 – Departmental Summary Report (Analytical Layer)

🎯 Objective

Using the previously loaded detailed employee table (TGT_EMPLOYEE_DETAIL), build a new mapping to summarize employee performance **by department**.

⚙️ Requirements

Source:

- Use TGT_EMPLOYEE_DETAIL (the result of Task 1) as your new source.

Transformations to Implement:

1. **Source Qualifier (SQ_EMP_DETAIL):**
    - Read all 72 records from the target of Task 1.
2. **Aggregator Transformation (AGG_DEPT_SUM):**
    - Group data by Department ID and Department Name.

- Create the following output columns:
    - EMP_COUNT → Count of employees per department
    - AVG_SALARY → Average of salary
    - AVG_TOTAL_INCOME → Average of total income
    - MAX_SALARY → Maximum salary in department
    - MIN_SALARY → Minimum salary in department
    - HIGH_SALARY_COUNT → Count of employees with grade = 'HIGH'

3. **Expression Transformation (EXP_RANK):**
    - Add one derived field:
        - Department Performance Level:
            - “Top” if AVG_TOTAL_INCOME > 12000
            - “Average” if between 8000 and 12000
            - “Low” otherwise
4. **Target:**
    - Create target table: **TGT_DEPT_SUMMARY** (Target query definition below)
    - Load one record per department.

Deliverables:

- Mapping: m_DEPT_SUMMARY_LOAD
- Session: s_DEPT_SUMMARY_LOAD
- Workflow: wf_DEPT_SUMMARY_LOAD
- Validation: Number of rows = number of departments (around 10–12).

◆ Bonus / Optional Enhancements

- Combine both tasks under a **master workflow** (wf_EMPLOYEE_PERFORMANCE_MASTER) that runs Task 1 then Task 2 automatically.

- 🧱 **Task 1 Target: TGT_EMPLOYEE_DETAIL**

```
CREATE TABLE TGT_EMPLOYEE_DETAIL (
  EMPLOYEE_ID      NUMBER(10)    NOT NULL,
  FULL_NAME        VARCHAR2(100),
  JOB_ID           VARCHAR2(10),
  JOB_TITLE        VARCHAR2(50),
  DEPARTMENT_ID    NUMBER(10),
  DEPARTMENT_NAME  VARCHAR2(100),
  CITY             VARCHAR2(50),
  STATE_PROVINCE   VARCHAR2(50),
  COUNTRY_ID       VARCHAR2(5),
  SALARY           NUMBER(10,2),
  COMMISSION_PCT   NUMBER(5,2),
  TOTAL_INCOME     NUMBER(12,2),
  YEARS_OF_SERVICE NUMBER(5,2),
  SALARY_GRADE     VARCHAR2(10),
  LOAD_DATE        DATE DEFAULT SYSDATE
);
```

🧩 Task 2 Target: TGT_DEPT_SUMMARY

```
CREATE TABLE TGT_DEPT_SUMMARY (
  DEPARTMENT_ID       NUMBER(10)    NOT NULL,
  DEPARTMENT_NAME     VARCHAR2(100),
  EMP_COUNT           NUMBER(5),
  AVG_SALARY          NUMBER(12,2),
  AVG_TOTAL_INCOME    NUMBER(12,2),
  MAX_SALARY          NUMBER(12,2),
  MIN_SALARY          NUMBER(12,2),
  HIGH_SALARY_COUNT   NUMBER(5),
  PERFORMANCE_LEVEL   VARCHAR2(20),
  LOAD_DATE           DATE DEFAULT SYSDATE
);
```