**Statement of Work (SoW) for HR Data Processing & Analysis**

**1.Project Overview**

This project focuses on processing and analyzing HR employee data. Using **Microsoft Fabric**, the project will generate insights through reports, enabling HR to make data-driven decisions.

The goal is to ingest, clean, transform, and aggregate HR data to provide key insights into employee churn trends. The project will be completed within 30 hours, using the HR data stored in a CSV file. The final output will be a dashboard showing analysis of HR employee data.

**2.Software & Tools Required**

|  |  |
| --- | --- |
| **Software** | **Purpose** |
| **Microsoft Fabric** | **Data processing, transformation and reporting** |
| **Python (PySpark)** | **Data transformation and cleaning in Microsoft fabric** |
| **Notebook (Fabric)** | Running Python scripts interactively |
| Power BI (Fabric) | Data visualization & reporting (if applicable) |

**3. Medallion Architecture Overview**

**This project follows the Medallion Architecture, which consists of four structured layers:**

**Create lakehouse for each layer.**

|  |  |  |
| --- | --- | --- |
| **Layer** | **Purpose** | **Storage Format** |
| Staging | Upload HR data (HR\_DATA.csv) into the staging lakehouse | CSV |
| Bronze | Copy data from staging and load it into Bronze Lakehouse (LH) | Parquet |
| Silver | Perform transformations like deduplication, data type changes, and remove null values on data in Bronze LH | Delta Table |
| Gold | Create dimension and fact tables using Silver Layer tables | Delta Table |

**4. Data Sources & Schema**

Raw Data (CSV File) → Staging Layer

The HR data will be sourced from the CSV file HR\_DATA.csv for processing in the Staging Layer.

Please download the source CSV file from below link.

[HR\_DATA.CSV](https://onedrive.live.com/edit?id=64BC0EB6DD6E03BD!sd32cf07988b244c0b49959212feb1578&resid=64BC0EB6DD6E03BD!sd32cf07988b244c0b49959212feb1578&cid=64bc0eb6dd6e03bd&ithint=file%2Cxlsx&redeem=aHR0cHM6Ly8xZHJ2Lm1zL3gvYy82NGJjMGViNmRkNmUwM2JkL0VYbndMTk95aU1CRXRKbFpJU19yRlhnQnhueWRWeVVpTTRIV21tNU16a2NHWnc_ZT1BV1BJSU8&migratedtospo=true&wdo=2)

Columns:

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| id | int | Unique Employee ID |
| first\_name | varchar | Employee's first name |
| last\_name | varchar | Employee's last name |
| birthdate | date | Employee's birthdate |
| gender | varchar | Employee's gender (Male/Female/Other) |
| race | varchar | Employee's race (Optional, based on dataset) |
| department | varchar | Department the employee works in |
| jobtitle | varchar | Job title of the employee |
| location | varchar | Location of the employee |
| hire\_date | date | Date when the employee was hired |
| termdate | date | Termination date (if applicable) |
| location\_city | varchar | City where the employee works |
| location\_state | varchar | State where the employee works |

**5. Staging Layer (Raw Data Storage)**

The Staging Layer stores the raw CSV data as-is, with no transformations applied. It will be stored in CSV format for direct access.

|  |  |
| --- | --- |
| **Staging File** | **Source** |
| hr\_data\_staging.csv | [HR\_DATA.CSV](https://onedrive.live.com/edit?id=64BC0EB6DD6E03BD!sd32cf07988b244c0b49959212feb1578&resid=64BC0EB6DD6E03BD!sd32cf07988b244c0b49959212feb1578&cid=64bc0eb6dd6e03bd&ithint=file%2Cxlsx&redeem=aHR0cHM6Ly8xZHJ2Lm1zL3gvYy82NGJjMGViNmRkNmUwM2JkL0VYbndMTk95aU1CRXRKbFpJU19yRlhnQnhueWRWeVVpTTRIV21tNU16a2NHWnc_ZT1BV1BJSU8&migratedtospo=true&wdo=2) |

**6. Bronze Layer: Data Storage and Transformation**

The Bronze Layer copies the data from the Staging Layer and stores it in Parquet format.

|  |  |
| --- | --- |
| **Bronze Table** | **Source** |
| hr\_data\_bronze.parquet | hr\_data\_staging.csv |

**7. Silver Layer: Data Cleaning, Transformation & Enrichment**

The Silver Layer prepares the data by applying necessary transformations to make it usable for analysis. This includes deduplication, correcting data types, and removing null values. The data will be stored as Delta Tables.

Be cautious when handling columns with null values, as some of these columns may still contain meaningful information despite being null.

|  |  |
| --- | --- |
| **Silver Table** | **Operations Performed** |
| hr\_data\_silver (Delta Table) | Deduplication, data type corrections, and removal of null values |

**8. Gold Layer: Data Aggregation for Reporting**

The Gold Layer will create dimension and fact tables using the transformed data from the Silver Layer. These tables will serve as the source for generating real-time reports and KPIs for the HR team.

|  |  |
| --- | --- |
| **Gold Tables** | **Purpose** |
| **DimEmployee** | Employee-specific data |
| **DimLocation** | Location-specific data |
| **DimDepartment** | Department-specific data |
| **FactEmployee** | Aggregated metrics and analytics for employees and departments |

**Gold Table Schema:**

**a. DimEmployee (Dimension Table for Employees)**

This table contains details about employees.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| employee\_id | int | Unique Employee ID (Primary Key) |
| first\_name | varchar | Employee's first name |
| last\_name | varchar | Employee's last name |
| birthdate | date | Employee's birthdate |
| gender | varchar | Employee's gender (Male/Female/Other) |
| race | varchar | Employee's race |
| hire\_date | date | Date of hire |
| termination\_date | date | Date of termination (if applicable) |
| full\_name | varchar | Concatenation of first and last name (for reporting purposes) |

**b.** **DimLocation (Dimension Table for Locations)**

This table contains details about the location of each employee’s job.

Give unique location id to each unique location\_city and location\_state combination according your choice.

Ex:

|  |  |  |
| --- | --- | --- |
| **Location id** | **Location\_city** | **Location\_state** |
| 101 | New York | New York |
| 102 | Los Angeles | Calofornia |

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| **location\_id** | int | Unique Location ID (Primary Key) |
| **location\_city** | varchar | City of the job location |
| **location\_state** | varchar | State of the job location |
| **location** | varchar | Full location description(loc\_city,loc\_state) |

**c. DimDepartment (Dimension Table for Departments)**

This table contains details about the departments where employees work.

Give unique **department\_id** manually to each unique **department** according your choice.

Give unique **department** id to each unique **department**  according your choice.

Ex:

|  |  |
| --- | --- |
| **Department\_ID** | **Department** |
| 1 | Engineering |
| 2 | HR |

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| department\_id | int | Unique Department ID (Primary Key) |
| department\_name | varchar | Department name |

**e. FactEmployee (Fact Table for Employee Metrics)**

This fact table contains aggregated data related to employees, including their department, location, and status, with the employee's ID as the key.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| fact\_id | int | Unique Fact ID (Primary Key) |
| employee\_id | int | Foreign Key from DimEmployee |
| location\_id | int | Foreign Key from DimLocation |
| department\_id | int | Foreign Key from DimDepartment |
| total\_employees | int | Total number of employees in each department |
| avg\_age | float | Average age of employees in the department (calculated from birthdate) |
| gender\_distribution | string | Distribution of gender per department |
| turnover\_rate | float | Employee turnover rate per department, calculated from the termdate |

**Step-by-Step Aggregation Logic:**

**Total Employees:**

Count the number of employees (id) per department.

**Average Age:**

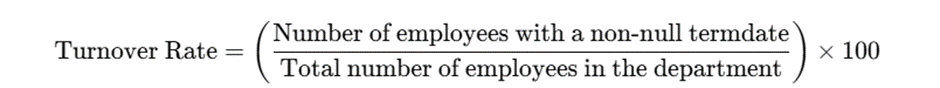
Calculate the average age of employees per department. This can be derived from the birthdate column.

**Gender Distribution:**

Calculate the percentage of male, female, and other employees per department.

**Turnover Rate:**

Calculate the turnover rate by dividing the number of employees with a non-null termdate by the total number of employees in the department.



**Location:**

Group by location\_city and location\_state to capture location-wise employee information.

**This is how your final FactEmployee table should look like:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **fact\_id** | **employee\_id** | **location\_id** | **department\_id** | **total\_employees** | **avg\_age** | **gender\_distribution** | **turnover\_rate** |
| 1 | 101 | 1 | 1 | 25 | 35.2 | Male: 40%, Female: 60% | 20.0 |
| 2 | 102 | 2 | 2 | 15 | 30.1 | Male: 50%, Female: 50% | 15.0 |
| 3 | 103 | 3 | 3 | 30 | 40.0 | Male: 70%, Female: 30% | 25.0 |

**9. Final Deliverables**

* Silver & Gold Layers in Delta Table format.
* Power BI Dashboard showing analysis of HR Employee data
* Project Documentation with detailed explanations of data transformations and report usage.

**10. Project Instructions**

* Follow the Medallion Architecture for data processing.
* Use proper naming conventions for all target files and Delta tables.
* Maintain clean and consistent coding standards in Python scripts and SQL queries for the Gold table.
* Automate the data pipeline.
* Implement proper auditing at each stage of the pipeline to track data processing.
* Ensure clear documentation of the entire project, including transformation steps and usage guidelines.
* Use version control (e.g., Git) to store all code, transformation scripts, and relevant files.
* Include the Git repository URL in the final documentation for submission.