Project Overview

Objective:

- Analyze and visualize HR data using Microsoft Fabric.
- Uncover insights on employee attrition, departmental trends, and termination date.
- Implement the Medallion Architecture: Staging → Bronze → Silver → Gold.

Tools & Technologies:

- Microsoft Fabric
- Dataflow
- PySpark (Notebooks)
- Power BI

Data Source:

- HR_DATA.csv containing employee profile information (name, age, gender, department, etc.).

Project Workflow:

- Ingest data into Staging Layer.
- Clean and move to Bronze Layer.
- Transform/enrich in Silver Layer.
- Analyze and aggregate in Gold Layer.
- Build visual dashboards in Power BI.

Staging & Bronze Layers

Staging Layer:

- Uploaded CSV to OneLake within Fabric workspace.
- Viewed raw data in Lakehouse "Files" tab.
- No transformations—used for lineage and backup.

Bronze Layer:

- Promoted headers and inferred schema.
- Validated data types and structure.
- Stored as structured Bronze Table

Silver Layer – Processing Summary

Step 1: Read Bronze Layer Data

- Loaded raw HR data from Bronze Layer (Parquet format).

Step 2: Data Profiling

- Counted null values in each column to assess data quality.

Step 3: Data Cleaning & Transformation

1. Removed dashes from 'id' column:
   - Created new column: int_id (cleaned version)
   - Example: "123-45" → 12345 (integer)
2. Converted date columns from string to DateType:
   - 'birthdate' and 'hire_date' parsed using format: M/d/yyyy
3. Renamed column:
   - int_id → employee_id

Step 4: Save to Silver Layer

- Saved cleaned HR data as Delta Table to OneLake:
- Path:

  abfss://FabricTrainingWorkspace@onelake.dfs.fabric.microsoft.com/silver_lakehouse_1.Lakehouse/Tables/hr_data_silver

  - Format: Delta
  - Mode: Overwrite

Step 5: Read Back for Verification

-Loaded the Delta table and displayed for visual confirmation.

Final Output Columns in Silver Layer:

Column Name	Data Type	Nullable	Description
id	string	true	Original employee ID with dashes
first_name	string	true	First name of the employee
last_name	string	true	Last name of the employee
birthdate	date	true	Converted to proper DateType
gender	string	true	Gender of the employee
race	string	true	Race/ethnicity of the employee
department	string	true	Department name
jobtitle	string	true	Job title
location	string	true	Office location
hire_date	date	true	Converted to proper DateType
termdate	timestamp	true	Termination date (if applicable)
location_city	string	true	City of work location
location_state	string	true	State of work location
employee_id	integer	true	Cleaned numeric employee ID (from `id`)

Documentation: DimEmployee Table Creation (Gold Layer)

Objective: Create a dimension table DimEmployee with detailed employee information.

Operations: Selected key columns, renamed id to employee_id, termdate to termination_date, and created full_name by combining first_name and last_name. Saved the result as a Delta table in the Gold Layer.

Output Columns: employee_id, first_name, last_name, birthdate, gender, race, hire_date, termination_date, full_name.

DimEmployee Table Creation (Gold Layer)

Objective: Create a dimension table DimEmployee with detailed employee information.

Operations: Selected key columns, renamed id to employee_id, termdate to termination_date, and created full_name by combining first_name and last_name. Saved the result as a Delta table in the Gold Layer.

Output Columns: employee_id, first_name, last_name, birthdate, gender, race, hire_date, termination_date, full_name.

DimDepartment Table Creation (Gold Layer)

Created the DimDepartment dimension table by selecting distinct departments from the silver layer and removing nulls. Assigned a unique department_id starting from 201 using row_number(). Saved the result as a Delta table in the Gold Lakehouse.

Output Columns: department_id, department

Fact Table Creation from HR Data - Documentation

Objective

To create a FactEmployee table that aggregates employee statistics such as total count, average age, gender distribution, and turnover rate based on department and location.

Steps Explained

1. Import Required PySpark Functions

Imports necessary functions for aggregation, transformation, and column generation.

2. Clean Gender Values

Standardizes the `gender` column by removing leading/trailing spaces and converting to lowercase for consistent processing.

3. Aggregate Employee Statistics

Group By: `department_id`, `location_id`

- total_employees: Count of employees in each group.
- avg_age: Rounded average age of employees in each group.
- gender_distribution: Shows gender composition as a formatted string (Male, Female, Other) in percentages.
- turnover_rate: Percentage of employees with a non-null termination date.

4. Generate Surrogate Keys

- fact_id: Unique identifier for each row in the fact table.
- employee_id: Synthetic identifier to simulate employee linkage (for dimensional modeling).

5. Select and Arrange Columns

Rearranges columns into a clean and logical order for storage and querying.

6. Display Final Fact Table

Displays the final transformed DataFrame containing employee statistics for review.

Output Schema

Column Name	Description
fact_id	Unique ID for each fact row
employee_id	Synthetic ID representing employee entity
location_id	Foreign key referencing location dimension
department_id	Foreign key referencing department dimension

total_employees	Count of employees in the department-location group
avg_age	Average age of employees
gender_distribution	Gender ratio in format: "Male: x%, Female: y%, Other: z%"
turnover_rate	Percentage of employees who left the organization