**EXTRACT, TRANSFORM AND LOAD**

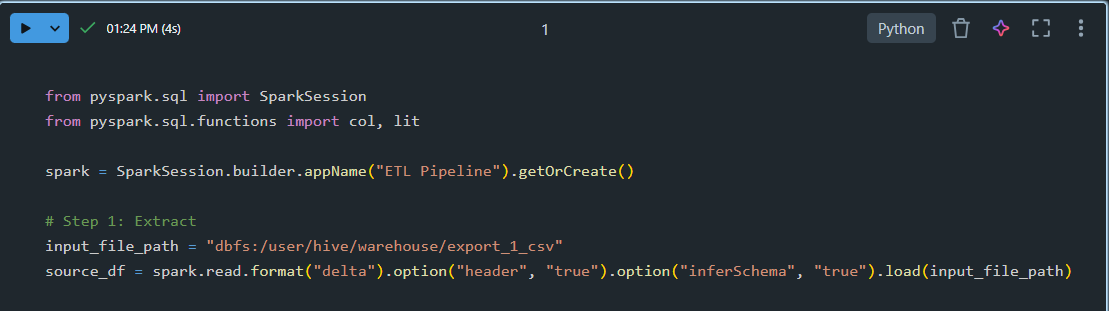
**HANDS-ON CASE STUDY**

**Name: S.Nikhil Sai**

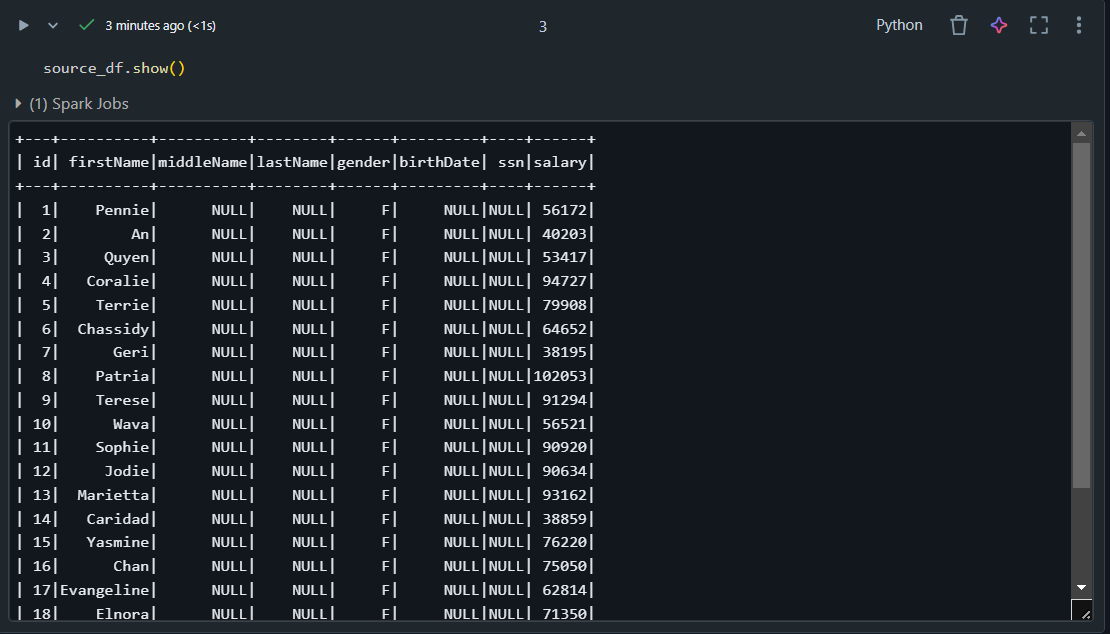
**Date: 8-12-2024**

**CREATING THE ETL NOTEBOOK:**

**STEP 1: Extract**

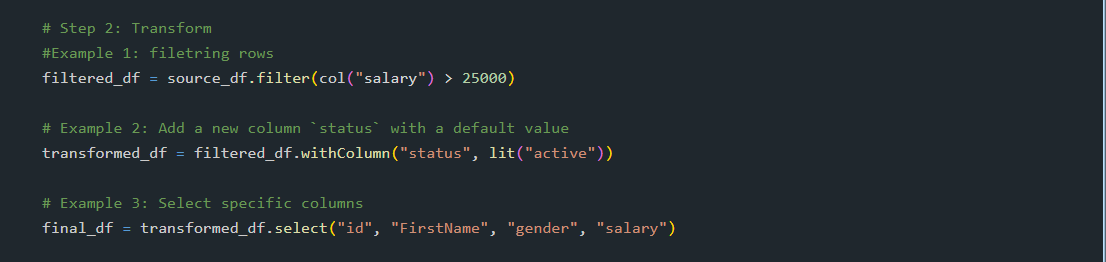
****

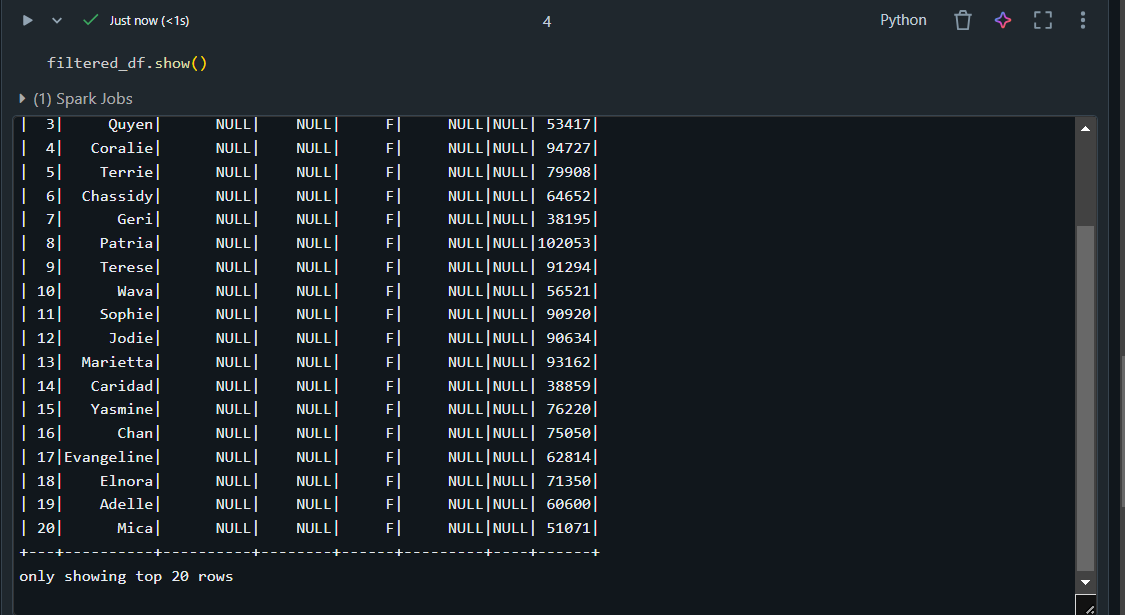
1. SparkSession:
   * This initializes the Spark application with the name "ETL Pipeline".
   * The getOrCreate() method either creates a new session or retrieves an existing one.
2. input\_file\_path:
   * Specifies the path to the data source file, located in the Databricks File System (DBFS).
   * The data is assumed to be in Delta format, a storage format optimized for big data operations.
3. spark.read.format("delta"):
   * Reads the data in Delta format.
   * Options:
     + "header": "true" indicates the file has a header row, so column names are inferred from the first row.
     + "inferSchema": "true" automatically determines the data types for each column.
4. load(input\_file\_path):
   * Loads the Delta file into a Spark DataFrame (source\_df) for further processing.



**STEP 2: Transform**

In this step, transformations are applied to the data to filter rows, add new columns, or reshape the data as needed.

****



* Retains only rows where the "salary" column is greater than 25,000.
* **col("salary"):** Refers to the salary column.
* **.filter:** Applies a filtering condition.

A screenshot of a computer

Description automatically generated

* Adds a new column named status with the value "active" for all rows.
* **withColumn:** Creates a new column or replaces an existing one.
* **lit("active"):** Creates a literal value "active" for the new column.

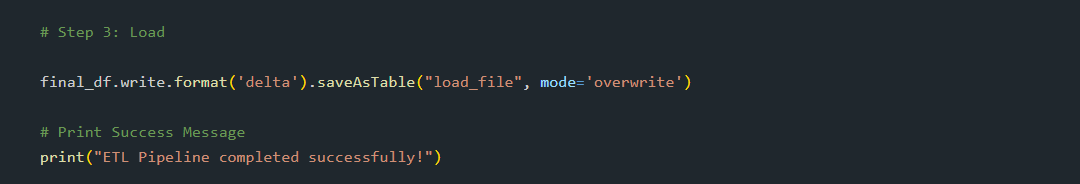
A screenshot of a computer

Description automatically generated

* Narrows down the DataFrame to only the specified columns: id, FirstName, gender, and salary.
* **.select:** Specifies the columns to retain in the output DataFrame.

**STEP 3: Load**

The final transformed DataFrame is written to a Delta table for storage.



A screenshot of a computer

Description automatically generated

* **final\_df.write:** Initiates the write operation for the DataFrame.
* **.format('delta'):** Specifies the output format as Delta.
* **.saveAsTable("load\_file", mode='overwrite'):**
* Saves the DataFrame as a table named "load\_file".
* mode='overwrite' ensures any existing data in the table is replaced.
* At the end of the pipeline, a success message is printed:

**RUNNING THE PIPELINE:**

**Step 1: create a new job.**

A screenshot of a computer

Description automatically generated

**Step 2: connect the etl file to the new job.**

A screenshot of a computer

Description automatically generated

**Step 3: connect the cluster.**

A screenshot of a computer

Description automatically generated

**Step 4: Now run the job.**

A screenshot of a computer

Description automatically generated

Therefore, the pipeline has successfully executed.