LAB 02: ELT Ingestion & Transformation

Duration: ~40 min | **Day:** 1 | **After module:** M02: ELT & Ingestion | **Difficulty:** Beginner-Intermediate

---

Scenario

> *"Load raw customer, order, and product data into the Bronze layer of RetailHub's data lakehouse. Apply transformations and save cleaned data as Delta tables."*

Objectives

After completing this lab you will be able to: - Read CSV/JSON with explicit schemas - Apply transformations (filter, withColumn, lower, trim) - Create temporary views and query with SQL - Save DataFrames as Delta tables - Verify results using SQL

Prerequisites

- Cluster running and attached to notebook
- Dataset files uploaded to Volume (from LAB 01)

Tasks Overview

Open **`LAB_02_code.ipynb`** and complete the `# TODO` cells.

Task	What to do	Key concept
Task 1	Read CSV with Explicit Schema	`StructType([StructField(...)])`, `.schema(schema)`
Task 2	Read JSON File	`spark.read.json(path)`
Task 3	Read Products CSV	`inferSchema=True`
Task 4	Transform Customer Data	`lower()`, `trim()`, `filter()`, `withColumn()`
Task 5	Create TempView + Query SQL	`.createOrReplaceTempView()`, `spark.sql()`
Task 6	Save as Delta Table	`.write.mode("overwrite").saveAsTable()`
Task 7	Verify with SQL	`SELECT COUNT(*)`, `DESCRIBE TABLE`

Detailed Hints

Task 1: Explicit Schema

- Define `StructType` with `StructField(name, type, nullable)`
- Common types: `StringType()`, `IntegerType()`, `DoubleType()`

Task 4: Transformations

- `lower(col("name"))` — convert to lowercase
- `trim(col("email"))` — remove leading/trailing spaces
- `filter(col("country") == "USA")` — filter rows

Task 5: TempView

- `df.createOrReplaceTempView("view_name")`
- Then: `spark.sql("SELECT * FROM view_name")`

Task 6: Save Delta

- `.write.mode("overwrite").saveAsTable("catalog.schema.table")`

Summary

In this lab you: - Read CSV/JSON data with explicit and inferred schemas - Applied column transformations (lower, trim, filter) - Created temporary views for SQL access - Saved clean data as managed Delta tables - Verified results using SQL queries

> **Exam Tip:** Explicit schemas are faster (no schema inference scan). `createOrReplaceTempView` is session-scoped. `saveAsTable` creates a managed Delta table in Unity Catalog.

> **What's next:** In LAB 03 you will work with Delta Lake operations — MERGE, UPDATE, DELETE, time travel, and RESTORE.