



Data Cleaning and Complex Transformations with Delta Lake





Agenda

- Introduction
- Agenda
- Handling Missing Values
- Removing Duplicate Data
- Data Type Corrections and Validation
- Joining Datasets
- Aggregations and Grouping Data
- Window Functions
- Hands-On Lab Overview

A blue pen is positioned diagonally across the center of the image, pointing towards the bottom right. The background is a light blue document featuring a bar chart with several vertical bars of varying heights. The overall image has a soft, slightly blurred aesthetic.

Strategies for Handling Missing Data

- Strategies for Handling Missing Values
 - Drop missing values
 - Fill missing values
 - Impute missing values
- PySpark Functions for Missing Values
 - `dropna()`
 - `fillna()`
 - `na.replace()`
- Importance of Domain Knowledge
 - Crucial for accurate data filling



Causes of Missing Data

Causes of Missing Data

Data entry errors

Non-response in surveys

Technical issues

Impact on Analysis

Skewed results

Reduced statistical power

Handling Techniques

Imputation methods

Using algorithms to predict missing values

Excluding missing data

Removing Duplicate Data

Causes of duplicates

- Ingestion retries

Identify duplicates

- Use `dropDuplicates()`

Precise de-duplication

- Utilize key columns

Impact on data accuracy

- Improves reporting

Data Type Corrections and Validation



Importance of Correct Data Types

Avoid errors in data processing
Ensure meaningful calculations



Using `cast()` and `when()` Functions

Safely fix data types
Prevent data type errors



Validating Data Ranges and Constraints

Ensure data falls within acceptable ranges
Maintain data integrity

Joining Datasets



Join Types

Inner Join
Left Join
Right Join
Full Outer Join



Use Cases and Examples

Combining data from different sources
Enriching analysis



Performance Considerations

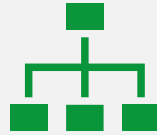
Efficiency with Spark
Optimization with Delta Lake

Aggregations and Grouping Data



Aggregate Functions

sum()
count()
avg()
max()
min()



Grouping Data

Using groupBy()



Use Cases

Sales totals by customer
Sales totals by region

Window Functions



Definition of Window Functions

Calculations across data partitions without aggregation



Common Examples

row_number()
rank()
dense_rank()
lag()
lead()



Use Cases

Ranking top customers
Running totals
Moving averages

Hands-On Lab Overview

- Clean messy sales data
 - Handle missing values
 - Remove duplicates
- Join sales with customer demographics
 - Combine sales data with demographic information
- Perform aggregations and window function calculations
 - Summarize data using aggregation functions
 - Use window functions for advanced calculations
- Validate with PySpark and SQL
 - Ensure data accuracy using PySpark
 - Use SQL for validation