**1. GroupBy and Aggregate Functions:**

* **Definition**: GroupBy is a method in PySpark used to group rows together based on one or more columns and perform aggregate functions on the grouped data.
* **Aggregate Functions:**
* **sum():** Computes the sum of values in a group.
* **min():** Finds the minimum value in a group.
* **max():** Finds the maximum value in a group.
* **avg():** Computes the average value in a group.
* **mean():** Computes the mean value in a group.
* **count():** Counts the number of rows in each group.

**2. Handling Missing Values:**

* **Dropping Rows with Null Values:**
* **na.drop():** Drops rows containing null values.
* **Parameters:**
* **how:** Specifies whether to drop rows if any or all values are null.
* **thresh:** Specifies the minimum number of non-null values required for a row to be kept.
* **subset:** Specifies the columns to consider for null value checks.
* Filling Missing Values:
* **na.fill():** Replaces null values with specified values.
* **Single Value:** Replace nulls with a single specified value.
* **Mean/Median/Mode:** Impute nulls with mean, median, or mode of the column using Imputer.

**3. Sorting and Ordering:**

* **Sorting:**
* **sort():** Sorts DataFrame based on specified columns.
* **orderBy():** Sorts DataFrame in ascending order based on specified columns.

**4. Joins:**

* **Definition:** Joins in PySpark are used to combine two DataFrames based on a common column(s).
* **Types of Joins:**
* **Inner Join:** Returns rows with matching keys in both DataFrames.
* **Outer Join:** Returns all rows when there is a match in either DataFrame.
* **Left Join:** Returns all rows from the left DataFrame and matching rows from the right DataFrame.
* **Right Join:** Returns all rows from the right DataFrame and matching rows from the left DataFrame.
* **Left Semi Join:** Returns rows from the left DataFrame for which a match exists in the right DataFrame.
* **Left Anti Join:** Returns rows from the left DataFrame for which no match exists in the right DataFrame.

**5. Union:**

**Definition:** Union is used to combine two DataFrames with the same schema.

**Distinct Union:** Use distinct() to merge without duplicates.

**6. User Defined Functions (UDF):**

* **Definition:** UDFs extend the functionality of PySpark by allowing users to define custom functions for DataFrame operations.
* **Creating UDF:**
* Define a Python function.
* Register the function as a UDF.
* **Applying UDF:**
* Use the UDF with withColumn() to apply it to DataFrame columns.
* Each topic provides essential functionalities and operations in PySpark for data manipulation, analysis, and transformation. Understanding and mastering these concepts are crucial for effective data processing in PySpark.