PySpark SQL provides various methods for DataFrame operations. Here's a list of some commonly used methods:

1. **select(*cols)**: Select specific columns from a DataFrame.
2. **filter(condition)**: Filter rows based on a condition.
3. **groupBy(*cols)**: Group rows based on one or more columns.
4. **agg(*expressions)**: Perform aggregate functions like sum, avg, max, min, etc.
5. **orderBy(*cols, **kwargs)**: Sort rows based on one or more columns.
6. **distinct()**: Get distinct rows from a DataFrame.
7. **join(other, on, how)**: Join two DataFrames based on a condition.
8. **union(other)**: Combine two DataFrames vertically.
9. **withColumn(colName, col)**: Add a new column or replace an existing one.
10. **alias(alias)**: Give a DataFrame an alias (alternate name).
11. **drop(*cols)**: Remove one or more columns from a DataFrame.
12. **describe(*cols)**: Compute summary statistics for specified columns.
13. **printSchema()**: Display the schema of the DataFrame.
14. **count()**: Count the number of rows in the DataFrame.
15. **show(*n, truncate=False)**: Display rows of the DataFrame.
16. **write.format(format).save(path)**: Write DataFrame to a file in a specified format.
17. **cache()**: Persist the DataFrame in memory for faster access.
18. **unpersist()**: Remove the DataFrame from memory.
19. **na.drop(how='any', subset=None)**: Remove rows with missing values.
20. **na.fill(value, subset=None)**: Fill missing values with a specified value.
21. **rollup(*cols)**: Perform rollup aggregation on specified columns.
22. **cube(*cols)**: Perform cube aggregation on specified columns.
23. **pivot(pivot_col, values=None)**: Pivot the DataFrame to create a pivot table.
24. **crossJoin(other)**: Perform a cross-join with another DataFrame.
25. **sample(withReplacement, fraction, seed)**: Randomly sample rows from the DataFrame.
26. **approxQuantile(col, probabilities, relativeError)**: Approximate quantiles of a column.
27. **stat**: Access statistical functions for the DataFrame.
28. **explain(extended=False)**: Display the execution plan of a DataFrame.

These are some of the most commonly used methods in PySpark SQL. Depending on your specific use case, you may also explore additional methods and functions provided by PySpark's rich ecosystem for data manipulation and analysis.