Command No.	Description	PySpark Command	Pandas Command
1	Load a CSV file	`df = spark.read.csv('file.csv')`	`df = pd.read_csv('file.csv')`
2	Display the first 5 rows	`df.show(5)`	`df.head(5)`
3	Print the schema	`df.printSchema()`	`df.dtypes`
4	Select a column	`df.select('column1')`	`df['column1']`
5	Filter rows	`df.filter(df['column1'] > 50)`	`df[df['column1'] > 50]`
6	Group by a column and count	`df.groupBy('column1').count()`	`df['column1'].value_counts()`
7	Sort by a column in descending order	`df.orderBy(df['column1'].desc())`	`df.sort_values('column1', ascending=False)`
8	Add a new column	`df.withColumn('new_column', df['column1'] * 2)`	`df['new_column'] = df['column1'] * 2`
9	Drop a column	`df.drop('column1')`	`df.drop('column1', axis=1)`
10	Count the number of rows	`df.count()`	`df.shape[0]`
11	Count distinct rows	`df.distinct().count()`	`df.nunique()`
12	Generate descriptive statistics	`df.describe().show()`	`df.describe()`
13	Fill NA/Null values	`df.fillna(value)`	`df.fillna(value)`
14	Apply a function to each row	`df.rdd.map(lambda x: (x, )).toDF()`	`df.apply(lambda x: function(x))`
15	Join two dataframes	`df.join(other_df, df['column1'] == other_df['column2'])`	`df.merge(other_df, left_on='column1', right_on='column2')`
16	Aggregate data	`df.agg({'column1': 'sum'})`	`df['column1'].sum()`
17	Rename a column	`df.withColumnRenamed('old_name', 'new_name')`	`df.rename(columns={'old_name': 'new_name'})`
18	Create a temporary view	`df.createOrReplaceTempView('table')`	`df.to_sql('table', con)`
19	Execute SQL query	`spark.sql('SELECT * FROM table')`	`pd.read_sql_query('SELECT * FROM table', con)`
20	Get number of partitions (PySpark) or shape of dataframe (Pandas)	`df.rdd.getNumPartitions()`	`df.shape`
21	Repartition dataframe (only in PySpark)	`df.repartition(5)`	N/A
22	Write dataframe to CSV	`df.write.csv('path')`	`df.to_csv('path')`
23	Write dataframe to Parquet	`df.write.parquet('path')`	`df.to_parquet('path')`
24	Write dataframe to JSON	`df.write.json('path')`	`df.to_json('path')`
25	Save dataframe as a table	`df.write.saveAsTable('table')`	`df.to_sql('table', con)`

Command No.	Description	PySpark Command	Pandas Command
26	Drop duplicates	`df.dropDuplicates()`	`df.drop_duplicates()`
27	Get column names	`df.columns`	`df.columns`
28	Set a column as index	N/A	`df.set_index('column1')`
29	Reset index	N/A	`df.reset_index()`
30	Get column data type	`df.dtypes`	`df.dtypes`
31	Change column data type	`df.withColumn("column1", df["column1"].cast(IntegerType()))`	`df['column1'].astype('int')`
32	Count nulls in column	`df.filter(df['column1'].isNull()).count()`	`df['column1'].isnull().sum()`
33	Replace nulls in column	`df.na.fill({'column1': 'value'})`	`df['column1'].fillna('value')`
34	Apply a function to a column	`df.withColumn('column1', func(df['column1']))`	`df['column1'].apply(func)`
35	Concatenate two columns	`df.withColumn('new_column', F.concat(df['column1'], df['column2']))`	`df['new_column'] = df['column1'] + df['column2']`
36	Split a column	`df.withColumn('split_column', F.split(df['column1'], 'delimiter'))`	`df['column1'].str.split('delimiter')`
37	Extract year from date	`df.withColumn('year', F.year(df['date_column']))`	`df['date_column'].dt.year`
38	Extract month from date	`df.withColumn('month', F.month(df['date_column']))`	`df['date_column'].dt.month`
39	Extract day from date	`df.withColumn('day', F.dayofmonth(df['date_column']))`	`df['date_column'].dt.day`
40	Get the current date	`df.withColumn('current_date', F.current_date())`	`pd.to_datetime('today')`
41	Add days to date	`df.withColumn('new_date', F.date_add(df['date_column'], 5))`	`df['date_column'] + pd.DateOffset(days=5)`
42	Subtract days from date	`df.withColumn('new_date', F.date_sub(df['date_column'], 5))`	`df['date_column'] - pd.DateOffset(days=5)`
43	Get the difference between two dates	`df.withColumn('date_diff', F.datediff(df['date1'], df['date2']))`	`(df['date1'] - df['date2']).dt.days`
44	Extract hour from timestamp	`df.withColumn('hour', F.hour(df['timestamp_column']))`	`df['timestamp_column'].dt.hour`