SQL & PySpark Equivalent

SQL	PySpark
SELECT	df.select()
DISTINCT	df.distinct()
WHERE	df.filter()
GROUP BY	df.groupBy()
HAVING	df.having()
ORDER BY	df.orderBy()
LIMIT	df.limit()
JOIN	df.join()

SQL	PySpark
UNION	df.union()
INTERSECT	df.intersect()
EXCEPT	df.except()
COUNT	df.groupBy().count()
SUM	df.groupBy().sum()
AVG	df.groupBy().avg()
MIN	df.groupBy().min()
MAX	df.groupBy().max()
COALESCE	df.na.fill()

SQL	PySpark
LIKE	df.filter(col.like(pattern))
IN	df.filter(col.isin([val1, val2, ...]))
BETWEEN	df.filter((col >= min_val) & (col <= max_val))
IS NULL	df.filter(col.isNull())
IS NOT NULL	df.filter(col.isNotNull())
SUBSTRING	df.selectExpr("SUBSTRING(column, start_index, length)") df.withColumn("new_column",
CONCAT	concat(col1, col2, ...)) df.withColumn("new_column",
TRIM	trim(col))

SQL	PySpark
UPPER	df.withColumn("new_column", upper(col))
LOWER	df.withColumn("new_column", lower(col))
ROUND	df.withColumn("new_column", round(col, num_decimals))
DATE_FORMAT	df.withColumn("new_column", date_format(col, "format"))
CAST	df.withColumn("new_column", col.cast("data_type"))
ROW_NUMBER()	row_number().over(Window.partitionBy().orderBy())

https://www.seekhobigdata.com/