15 FUNCTIONS EXCEL SQL & PYTHON

→

SUM

Excel: SUM()

SQL: SUM()

Python*: df['column'].sum()

*using pandas

AVERAGE

Excel: AVERAGE()

SQL: AVG()

Python*: df['column'].mean()

*using pandas

COUNT

Excel: COUNT()

SQL: COUNT()

Python*: df['column'].count()

*using pandas

MAX

Excel: MAX()

SQL: MAX()

Python: df['column'].max()

MIN

Excel: MIN()

SQL: MIN()

Python*: df['column'].min()

*using pandas

IF

Excel: IF()

SQL: CASE WHEN...THEN...

Python*:df['column'].apply(lambda x: ...)

*using pandas

LEFT JOIN

Excel: VLOOKUP()

SQL: LEFT JOIN...ON...

Python*:df1.merge(df2, on='key', how='left)

*using pandas

OTHER JOINS

Excel: HLOOKUP()

SQL: N/A (use JOINs)

Python*:df1.T.merge(df2.T, on='key', how='left').T

*using pandas

CONCATENATE

Excel: CONCATENATE()

SQL: CONCAT()

Python: Use the '+' operator

LEFT

Excel: LEFT()

SQL: LEFT()

Python*:df['column'].str[:n]

*using pandas

RIGHT

Excel: RIGHT()

SQL: RIGHT()

Python:df['column'].str[-n:]

MIDDLE

Excel: MID()

SQL: SUBSTRING()

Python:df['column'].str[start:end]

LENGTH

Excel: LEN()

SQL: LENGTH()

Python:df['column'].str.len()

CONVERT TO TEXT

Excel: TEXT()

SQL: FORMAT()

Python:df['column'].apply(lambda x: f'{x:.2f}')

Remember to save this aid for future reference and share it with your fellow data analysts!

If you found this helpful, make sure to hit that "Like" button