import pandas as pd

import sqlite3 # or use `psycopg2` for PostgreSQL / `mysql.connector` for MySQL

# CONFIGURATIONS

csv\_file\_path = "your\_input\_file.csv"

database\_path = "reports.db" # SQLite DB file

db\_table\_name = "report\_fields"

report\_column\_name = "report" # Change if your CSV has a different header

# 1. Read CSV File

df = pd.read\_csv(csv\_file\_path)

# 2. Identify field columns (exclude path/report name)

field\_columns = [col for col in df.columns if col.lower() not in ['path', report\_column\_name.lower()]]

# 3. Prepare list of rows for DB insertion

data\_rows = []

pk = 1

for \_, row in df.iterrows():

report\_name = row[report\_column\_name]

for col in field\_columns:

value = str(row[col]).strip().lower()

if value in ['a', 'ab', 'b']:

# Convert field name from table1\_field1 → table1.field1

table\_field = col.replace('\_', '.', 1)

data\_rows.append((pk, report\_name, table\_field, value))

pk += 1

# 4. Connect to DB and create table if not exists

conn = sqlite3.connect(database\_path)

cursor = conn.cursor()

cursor.execute(f"""

CREATE TABLE IF NOT EXISTS {db\_table\_name} (

id INTEGER PRIMARY KEY,

report\_name TEXT NOT NULL,

field\_name TEXT NOT NULL,

marker TEXT CHECK(marker IN ('a', 'ab', 'b')) NOT NULL

);

""")

# 5. Insert data into table

cursor.executemany(f"""

INSERT INTO {db\_table\_name} (id, report\_name, field\_name, marker)

VALUES (?, ?, ?, ?);

""", data\_rows)

conn.commit()

conn.close()

print("✅ Data successfully inserted into the database.")

import pandas as pd

# Step 1: Load Excel

Df = pd.read\_excel(“your\_input\_file.xlsx”)

# Step 2: Extract relevant columns

Report\_col = ‘Report’ # Change if your report column header is different

Field\_cols = [col for col in df.columns if col not in [‘Path’, report\_col]]

# Step 3: Prepare output rows

Output\_rows = []

Pk = 1

For \_, row in df.iterrows():

Report\_name = row[report\_col]

For col in field\_cols:

Value = str(row[col]).strip().lower()

If value in [‘a’, ‘ab’, ‘b’]:

Table\_field = col.replace(‘\_’, ‘.’, 1) # Only replace first underscore

Output\_rows.append([pk, report\_name, table\_field, value])

Pk += 1

# Step 4: Create final DataFrame

Output\_df = pd.DataFrame(output\_rows, columns=[‘ID’, ‘Report Name’, ‘Field Name’, ‘Marker’])

# Step 5: Save to CSV/Excel if needed

Output\_df.to\_csv(“transformed\_report\_fields.csv”, index=False)

# output\_df.to\_excel(“transformed\_report\_fields.xlsx”, index=False)

Print(“✅ Transformation complete.”)