import pandas as pd

import tkinter as tk

from tkinter import filedialog, simpledialog, messagebox

import oracledb

import os

def connect\_to\_oracle(host, port, service\_name, username, password):

try:

dsn = f"{host}:{port}/{service\_name}"

conn = oracledb.connect(user=username, password=password, dsn=dsn, encoding="UTF-8")

return conn

except Exception as e:

messagebox.showerror("Connection Error", f"Failed to connect to Oracle DB:\n{e}")

return None

def get\_customer\_ref(conn, user\_id):

cursor = conn.cursor()

cursor.execute("""

SELECT customer\_ref FROM v\_nonvzw\_customer WHERE userid = :user\_id

""", [user\_id])

result = cursor.fetchone()

return result[0] if result else None

def get\_latest\_subscription(conn, customer\_ref):

cursor = conn.cursor()

cursor.execute("""

SELECT start\_date, end\_date

FROM (

SELECT start\_date, end\_date

FROM scm\_subscription

WHERE customer\_number = :customer\_number

ORDER BY

CASE

WHEN end\_date IS NULL THEN 1

ELSE 0

END DESC,

end\_date DESC

) WHERE ROWNUM = 1

""", [customer\_ref])

result = cursor.fetchone()

return result if result else (None, None)

def process\_file(file\_path, user\_id\_column, conn):

is\_excel = file\_path.endswith(".xlsx")

df = pd.read\_excel(file\_path) if is\_excel else pd.read\_csv(file\_path)

# Output lists

status\_list = []

start\_date\_list = []

end\_date\_list = []

for user\_id in df[user\_id\_column]:

customer\_ref = get\_customer\_ref(conn, user\_id)

if customer\_ref:

start\_date, end\_date = get\_latest\_subscription(conn, customer\_ref)

if start\_date:

start\_date\_list.append(start\_date)

else:

start\_date\_list.append(None)

if end\_date:

end\_date\_list.append(end\_date)

status\_list.append("Subscription Canceled")

else:

end\_date\_list.append(None)

status\_list.append("Subscription Active")

else:

start\_date\_list.append(None)

end\_date\_list.append(None)

status\_list.append("Customer Ref Not Found")

# Add to DataFrame

df["Subscription\_Status"] = status\_list

df["Subscription\_Start\_Date"] = start\_date\_list

df["Subscription\_End\_Date"] = end\_date\_list

# Save

output\_file = os.path.splitext(file\_path)[0] + "\_updated." + ("xlsx" if is\_excel else "csv")

if is\_excel:

df.to\_excel(output\_file, index=False)

else:

df.to\_csv(output\_file, index=False)

messagebox.showinfo("Success", f"Processed file saved as:\n{output\_file}")

def gui\_app():

root = tk.Tk()

root.withdraw()

# Step 1: Ask for DB connection info

host = simpledialog.askstring("Oracle Host", "Enter Oracle DB Host (e.g., localhost)")

port = simpledialog.askstring("Oracle Port", "Enter Oracle Port (e.g., 1521)")

service\_name = simpledialog.askstring("Service Name", "Enter Oracle Service Name (e.g., orclpdb1)")

username = simpledialog.askstring("Username", "Enter DB Username")

password = simpledialog.askstring("Password", "Enter DB Password", show='\*')

if not all([host, port, service\_name, username, password]):

messagebox.showerror("Error", "All DB connection fields are required.")

return

conn = connect\_to\_oracle(host, port, service\_name, username, password)

if not conn:

return # Connection failed

# Step 2: Upload file

file\_path = filedialog.askopenfilename(

title="Select Excel or CSV File",

filetypes=[("Excel files", "\*.xlsx"), ("CSV files", "\*.csv")]

)

if not file\_path:

messagebox.showerror("Error", "No file selected.")

return

# Step 3: Ask for user\_id column

is\_excel = file\_path.endswith(".xlsx")

df = pd.read\_excel(file\_path) if is\_excel else pd.read\_csv(file\_path)

column\_names = df.columns.tolist()

user\_id\_column = simpledialog.askstring(

"Column Name", f"Enter the user\_id column name from:\n{column\_names}"

)

if user\_id\_column not in column\_names:

messagebox.showerror("Error", f"Column '{user\_id\_column}' not found.")

return

# Step 4: Process

process\_file(file\_path, user\_id\_column, conn)

conn.close()

if \_\_name\_\_ == "\_\_main\_\_":

gui\_app()