In [4]:
```
import os
import sqlite3
import requests
import gzip
import xml.etree.ElementTree as ET

# Define namespaces
namespaces = {
    'tmx': 'http://www.lisa.org/tmx14'
}

# Define the TMX file URL
tmx_url = "https://opus.nlpl.eu/download.php?f=UN/v20090831/tmx/ar-en.tmx.gz"
tmx_file_path = "ar-en.tmx.gz"

# Download the TMX file
def download_tmx_file(url, file_path):
    response = requests.get(url)
    with open(file_path, 'wb') as f:
        f.write(response.content)

# Extract and transform TMX data
def transform_tmx_data(file_path):
    translations = []
    with gzip.open(file_path, 'rt', encoding='utf-8') as f:
        tree = ET.parse(f)
        root = tree.getroot()

        for tu in root.findall(".//tmx:tu", namespaces):
            source = tu.find(".//tmx:tuv[@xml:lang='en']/tmx:seg", namespaces)
            target = tu.find(".//tmx:tuv[@xml:lang='ar']/tmx:seg", namespaces)
            translations.append((source, target))

    return translations

# Load data into SQLite database
def load_data_into_database(data, database_path):
    conn = sqlite3.connect(database_path)
    cursor = conn.cursor()

    cursor.execute('''CREATE TABLE IF NOT EXISTS translations (source TEXT, ta

    cursor.executemany("INSERT INTO translations (source, target) VALUES (?, ?

    conn.commit()
    conn.close()

# Main function
def main():
    # Step 1: Download the TMX file
    if not os.path.exists(tmx_file_path):
        print("Downloading the TMX file...")
        download_tmx_file(tmx_url, tmx_file_path)

    # Step 2: Transform TMX data
    print("Transforming TMX data...")
    transformed_data = transform_tmx_data(tmx_file_path)
```

```
    # Step 3: Load data into the database
    database_path = "translations.db"
    print("Loading data into the database...")
    load_data_into_database(transformed_data, database_path)

    print("ETL process completed successfully!")

if __name__ == "__main__":
    main()
```

```
Transforming TMX data...
Loading data into the database...
ETL process completed successfully!
```

In [ ]: