```python
import logging
import re
import sqlite3
from time import time
import pandas as pd

def parse_line(line,
               ignore_pattern=r'^[%#].*',
               pattern=r'^(.+) (.+) \[([^\]]+)\] (/.+/)'):

    try:
        match = re.match(pattern, line)
        if match:
            traditional = match.group(1)
            simplified = match.group(2)
            pinyin = match.group(3)
            english = match.group(4).strip('/')

            return {
                'traditional': traditional,
                'simplified': simplified,
                'pinyin': pinyin,
                'english': english
            }

        else:
            raise ValueError(f'Failed to match expected pattern:\n\t {line}')
    except Exception as e:
        logging.error(f'{str(e)}')
        return None

def load_to_sqlite(filename, table_name, db_file, max_line=-1):
    con = sqlite3.connect(db_file)
    cur = con.cursor()

    # cur.execute(f'DROP TABLE IF EXISTS {table_name}')
    cur.execute(f'delete from {table_name}; ')
    cur.execute(f'''
            CREATE TABLE if not exists {table_name} (
                traditional TEXT,
                simplified TEXT,
                pinyin TEXT,
                english TEXT
            );
        ''')

    nline = 0
    with open(filename, 'r', encoding='utf-8') as f:
        for line in f:
            if max_line > 0 and nline > max_line:
                break

            if line.startswith('#') or line.startswith('%'):
                continue

            entry = parse_line(line)
            if entry is None:
                continue

            nline += 1
            values = (
                entry['traditional'],
                entry['simplified'],
                entry['pinyin'],
                entry['english'] )

            cur.execute(f'INSERT INTO {table_name} VALUES (?, ?, ?, ?);', values)
```

```
68
69
70          con.commit()
71          con.close()
72
73      if __name__ == '__main__':
74          start_ts = time()
75          filename, table_name, db_file = 'cedict_ts.u8', 't_mdbg_dict', 'cc_cedict.db'
76          load_to_sqlite(filename, table_name, db_file, max_line=1000)
77          end_ts = time()
78          print(f"Completed loading file '{filename}' into sqlite db table {table_name}' in {
            end_ts-start_ts} sec")
79
80          # verify
81          with sqlite3.connect(db_file) as con:
82              df = pd.read_sql(f"select count(*) from {table_name}", con)
83              print(df.head())
```