```
#!/usr/bin/env python3
########################################
### Assignment 5 - Visualizing Data
### Author: Jeremiah Purba
########################################

#imports at top of file
import sqlite3
import os
import logging
import matplotlib.pyplot as plt

def debug_config():
    logging.basicConfig(
        level=logging.ERROR,
        format = "[Degrees]:%(asctime)s:%(levelname)s:%(message)s"
    )   #DEBUG,INFO,ERROR,WARNING,CRITICAL

def db_checkfile(dbfile):
    if os.path.exists(dbfile) and os.path.getsize(dbfile) > 0:
        logging.debug("{a} found and not zero size".format(a=dbfile))
    else:
        logging.error("{a} found or not zero size".format(a=dbfile))

def db_connect(dbfile):
    con = sqlite3.connect(dbfile)
    logging.debug("DB Connected".format())
    return con

def db_cursor(con):
    cur = con.cursor()
    logging.debug("Cursor set".format())
    return cur

def db_runquery(cur,query):
    cur.execute(query)
    result = cur.fetchall()
    logging.debug("DB Query executed and returned".format())
    return result

def print_degrees(res):
    allyears = []
    HealthProfessions = []
    Education = []
    ComputerScience = []
    Engineering = []

    for degree in res:
        allyears.append(degree[0])
        HealthProfessions.append(degree[1])
        Education.append(degree[2])
        ComputerScience.append(degree[3])
        Engineering.append(degree[4])

    plt.plot(allyears, HealthProfessions, label='HealthProfessions')
    plt.plot(allyears, Education, label='Education')
    plt.plot(allyears, ComputerScience, label='ComputerScience')
    plt.plot(allyears, Engineering, label='Engineering')
    plt.xlabel('Year')
    plt.ylabel('Degrees')
    plt.title("""% of Bachelor's degrees for USA women by major (1970 -
2011)\nDegrees Over Time""")
    plt.legend()
    plt.show()

def main():
    dbfile = 'degrees2.db'
    programname = "Visualizing Data"
    debug_config()

    print(programname)
    db_checkfile(dbfile)
    try:
        con = db_connect(dbfile)
        cur = db_cursor(con)
        # Define the SELECT query to retrieve multiple column
        column_names = ['Year','HealthProfessions', 'Education', 'ComputerScience',
'Engineering']
        query2 = f"SELECT {', '.join(column_names)} FROM degrees"
        res = db_runquery(cur, query2)
        print_degrees(res)
    except sqlite3.Error as error:
        logging.error("Error executing query", error)
    finally:
        if con:
            con.close()
            logging.debug("[Info] DB Closed".format())

    print('Done - completed')
    logging.info("Completed.")

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