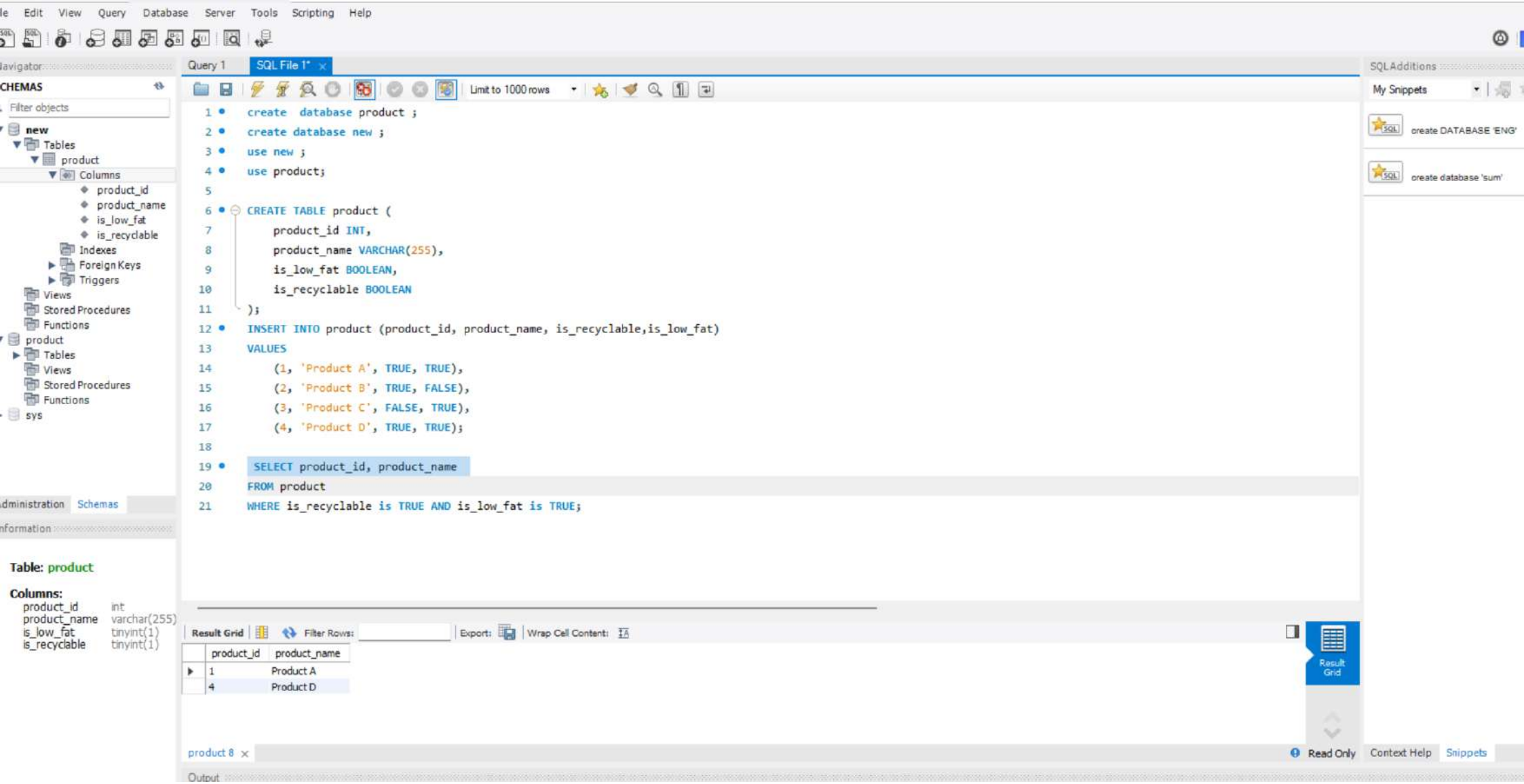
```python
import csv
import os

USER_DATA_FILE = "data.csv"
COLUMNS = ["username", "email", "password"]

if not os.path.exists(USER_DATA_FILE):
    with open(USER_DATA_FILE, "w") as file:
        writer = csv.DictWriter(file, fieldnames=COLUMNS)
        writer.writeheader()


def login():
    email = input("Email: ").strip().lower()
    password = input("Password: ").strip()

    with open(USER_DATA_FILE, "r") as file:
        users = csv.DictReader(file)
        for row in users:
            if row["email"] == email and row["password"] == password:
                print(f"Welcome back, {row['username']}!")
                return

    print("The email or password  is incorrect. Please try again .")



def register():
    username = input("Username: ").strip()
    email = input("Email: ").strip().lower()
    password = input("Password: ").strip()

    with open(USER_DATA_FILE, "r") as file:
        users = csv.DictReader(file)
        for row in users:
            if row["username"] == username:
```

```
28  def register():
39              if row["email"] == email:
40                  print("This email address is already exist ")
41                  return
42
43      with open(USER_DATA_FILE, "a") as file:
44          writer = csv.writer(file)
45          writer.writerow([username, email, password])
46
47  print("Account created successfully ")
48
49
50
51  def main():
52      while True:
53          print("\nWelcome to the authentication system!")
54          action = input("Would you like to register or log in? : ").strip().lower()
55
56          if action == "register":
57              register()
58          elif action == "login":
59              login()
60          else:
61              print("Invalid option. Please choose 'register' or 'login'.")
62
63
64  if __name__ == "__main__":
65      main()
66
```


le Edit View Query Database Server Tools Scripting Help
Navigator
CHEMAS
Filter objects
new
Tables
product
Columns
product_id
product_name
is_low_fat
is_recyclable
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
product
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
nformation
Table: product
Columns:
product_id int
product_name varchar(255)
is_low_fat tinyint(1)
is_recyclable tinyint(1)
Query 1
SQL File 1*
Limit to 1000 rows
1 create database product ;
2 create database new ;
3 use new ;
4 use product;
5
6 CREATE TABLE product (
7 product_id INT,
8 product_name VARCHAR(255),
9 is_low_fat BOOLEAN,
10 is_recyclable BOOLEAN
11);
12 INSERT INTO product (product_id, product_name, is_recyclable,is_low_fat)
13 VALUES
14 (1, 'Product A', TRUE, TRUE),
15 (2, 'Product B', TRUE, FALSE),
16 (3, 'Product C', FALSE, TRUE),
17 (4, 'Product D', TRUE, TRUE);
18
19 SELECT product_id, product_name
20 FROM product
21 WHERE is_recyclable is TRUE AND is_low_fat is TRUE;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
product_id product_name
1 Product A
4 Product D
Result Grid
product 8
Read Only
Context Help
Snippets
Output
SQLAdditions
My Snippets
create DATABASE 'ENG'
create database 'sum'

```python
def apply_operations(operations):
    result = 0

    for op in operations:
        if op == "++":
            result += 1
        elif op == "--":
            result -= 1

    return result

operations = ["++", "++", "--", "++"]
print(apply_operations(operations))
```