Section 2: Rest API.

a. Create a database (use sqlite3 database if using Python or any local DB) containing two tables Books and Authors. Add appropriate columns to the tables. Every book has a single author and referential integrity should be maintained
b. Create a Rest API which supports the following operations:
    i. Insert, update and select on Books

Handle exception for adding Authors who are not present in the table.

SOFTWARE SPECIFICATION

Operating System : Microsoft Windows 10 (64 bit)
Languages : Python (3.8.2)
Development Environment : JetBrain PyCharm Community Edition 2019.1.3

Databse: Sqlite 3

2.2.1 Description of Use Case

- Add book id and name in book table
- Update book name in Book Table
- View the entire book in Book Table
- Add author id and name in author table
- Add author id to book id in BookAuthor table

Source Code:

Database Creation:

db.py

```
import sqlite3

conn = sqlite3.connect("datagrokr.db")

print("Opened database successfully")

conn.execute('CREATE TABLE books(id INT ,title VARCHAR(255),PRIMARY KEY(id))')

conn.execute("CREATE TABLE Authors(id INT ,author_name VARCHAR(255),PRIMARY
KEY(id))")

conn.execute("CREATE TABLE BookAuthors(book_id int not null,author_id int not null,"
        "foreign key(author_id)references authors(id),foreign key(book_id)references
```

```
books(id))")

print("Table created Successfully")

conn.close()
```

app.py

```
from flask import Flask, render_template, request

import sqlite3 as sql

app = Flask(__name__, template_folder='template')


@app.route('/')
def home():
   return render_template('home.html')


@app.route('/enternew')
def enternew():
   return render_template("newbook.html")


@app.route('/updatebook')
def update_book():
   return render_template("updateinfo.html")


@app.route('/choosebook')
def selectbook():
   return render_template("choosebook.html")



@app.route('/addauthor')
def addauthor():
   return render_template("addauthor.html")



@app.route('/addbookauthor')
def addbookauthor():
   return render_template("addbookauthor.html")
```

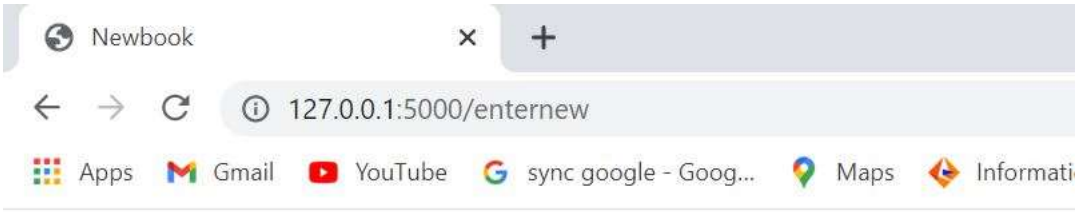
```
@app.route('/addbook', methods=['POST'])
def addbook():
   if request.method == 'POST':
      try:
         id = request.form['id']
         title = request.form['title']

         with sql.connect("datagrokr.db") as con:

            cur = con.cursor()

            cur.execute("INSERT INTO books VALUES(?,?)", (id, title))

            msg = "Record inserted successfully"

      except:
         con.rollback()
         msg = "Error while inserting"

      finally:
         return render_template("result.html", msg=msg)
         con.close()


@app.route('/updateinfo', methods=['POST', 'GET'])
def updateinfo():
   if request.method == 'POST':
      try:
         id = request.form['id']
         title = request.form['title']

         with sql.connect("datagrokr.db") as con:

            cur = con.cursor()

            cur.execute("update books set title=? where id=?", (title, id))

            msg = "Record updated successfully"

      except:
         con.rollback()
         msg = "Error while updating"
```

```
        finally:
            return render_template("result.html", msg=msg)
            con.close()


@app.route('/list')
def list():
    con = sql.connect("datagrokr.db")
    con.row_factory = sql.Row

    cur = con.cursor()

    cur.execute("select * from books")

    rows = cur.fetchall()

    return render_template("list.html", rows=rows)


@app.route('/addauthor', methods=['POST'])
def add_author():
    if request.method == 'POST':
        try:
            id = request.form['ida']
            author_name = request.form['author_name']

            with sql.connect("datagrokr.db") as con:

                cur = con.cursor()

                cur.execute("INSERT INTO authors VALUES(?,?)", (id, author_name))

                msg = "Record inserted successfully"

        except:
            con.rollback()
            msg = "Error while inserting"

        finally:
            return render_template("result.html", msg=msg)
            con.close()



@app.route('/addbookauthor', methods=['POST'])
def add_ids():
```

```
if request.method == 'POST':
    try:
        book_id = request.form['idb']
        author_id = request.form['ida']

        with sql.connect("datagrokr.db") as con:

            cur = con.cursor()

            cur.execute("INSERT INTO bookauthors VALUES(?,?)", (book_id, author_id))

            msg = "Record inserted successfully"

    except:
        con.rollback()
        msg = "Error while inserting"

    finally:
        return render_template("result.html", msg=msg)
        con.close()
```

Output:

```
D:\datagrokr.survey\venv\Scripts\python.exe D:/datagrokr.survey/library/app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 483-837-752
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

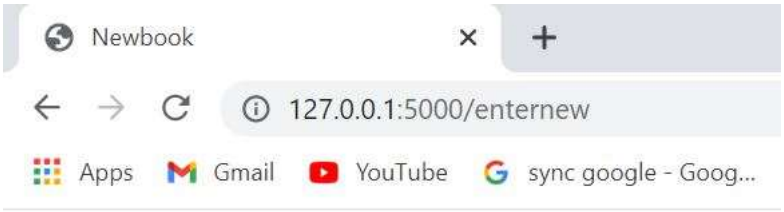


Add Book

Id

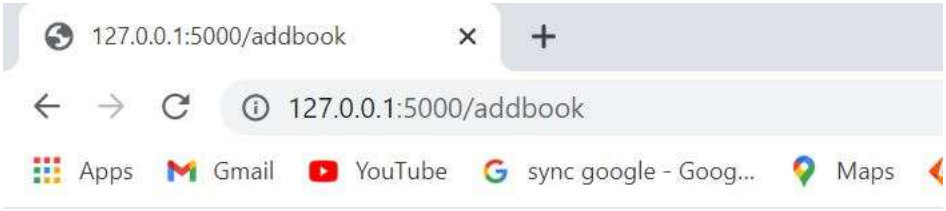
Title

submit



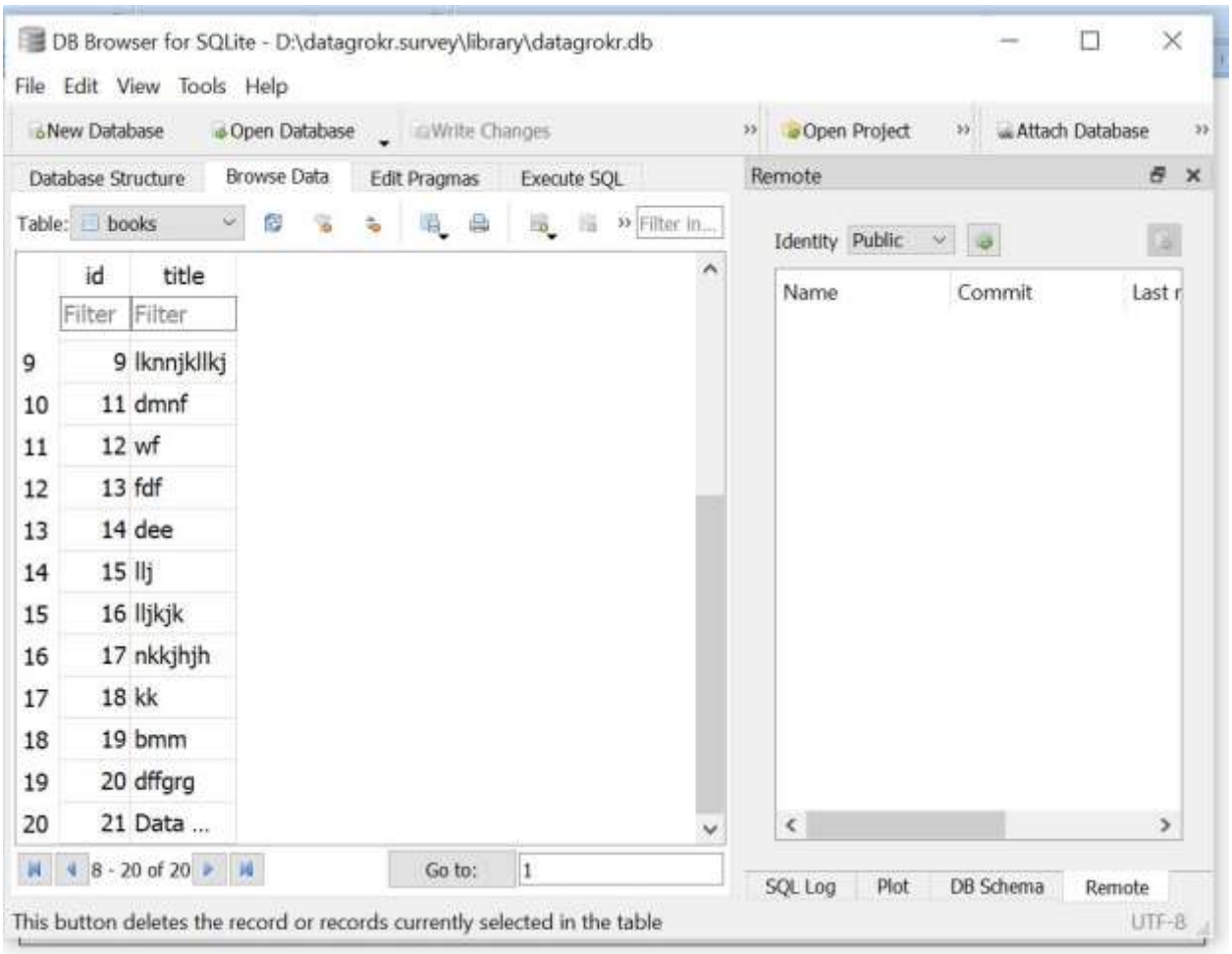
Add Book

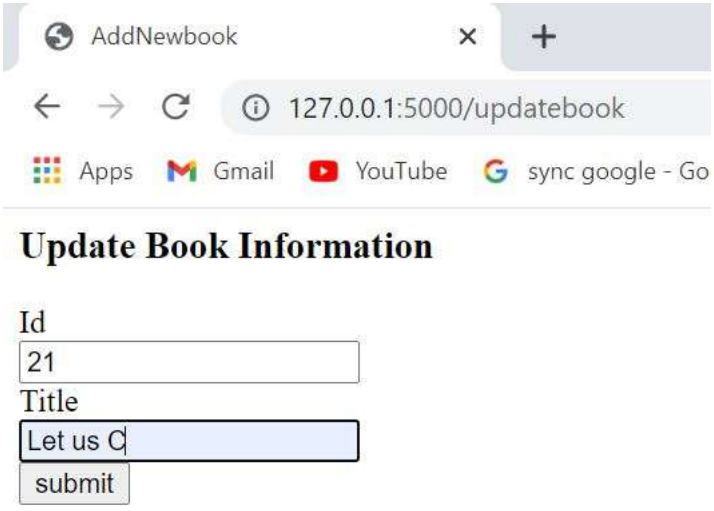
Id
21
Title
Data Structure
submit



Record inserted successfully

Go to home


DB Browser for SQLite - D:\datagrokr.survey\library\datagrokr.db
File Edit View Tools Help
New Database
Open Database
Write Changes
Open Project
Attach Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: books
Filter in...
id
title
Filter
Filter
9 9 lknnjkllkj
10 11 dmnf
11 12 wf
12 13 fdf
13 14 dee
14 15 llj
15 16 lljkjk
16 17 nkkjhjh
17 18 kk
18 19 bmm
19 20 dffgrg
20 21 Data ...
8 - 20 of 20
Go to:
1
Remote
Identity
Public
Name
Commit
Last r
SQL Log
Plot
DB Schema
Remote
This button deletes the record or records currently selected in the table
UTF-8


AddNewbook
127.0.0.1:5000/updatebook
Apps
Gmail
YouTube
sync google - Go
Update Book Information
Id
21
Title
Let us C
submit

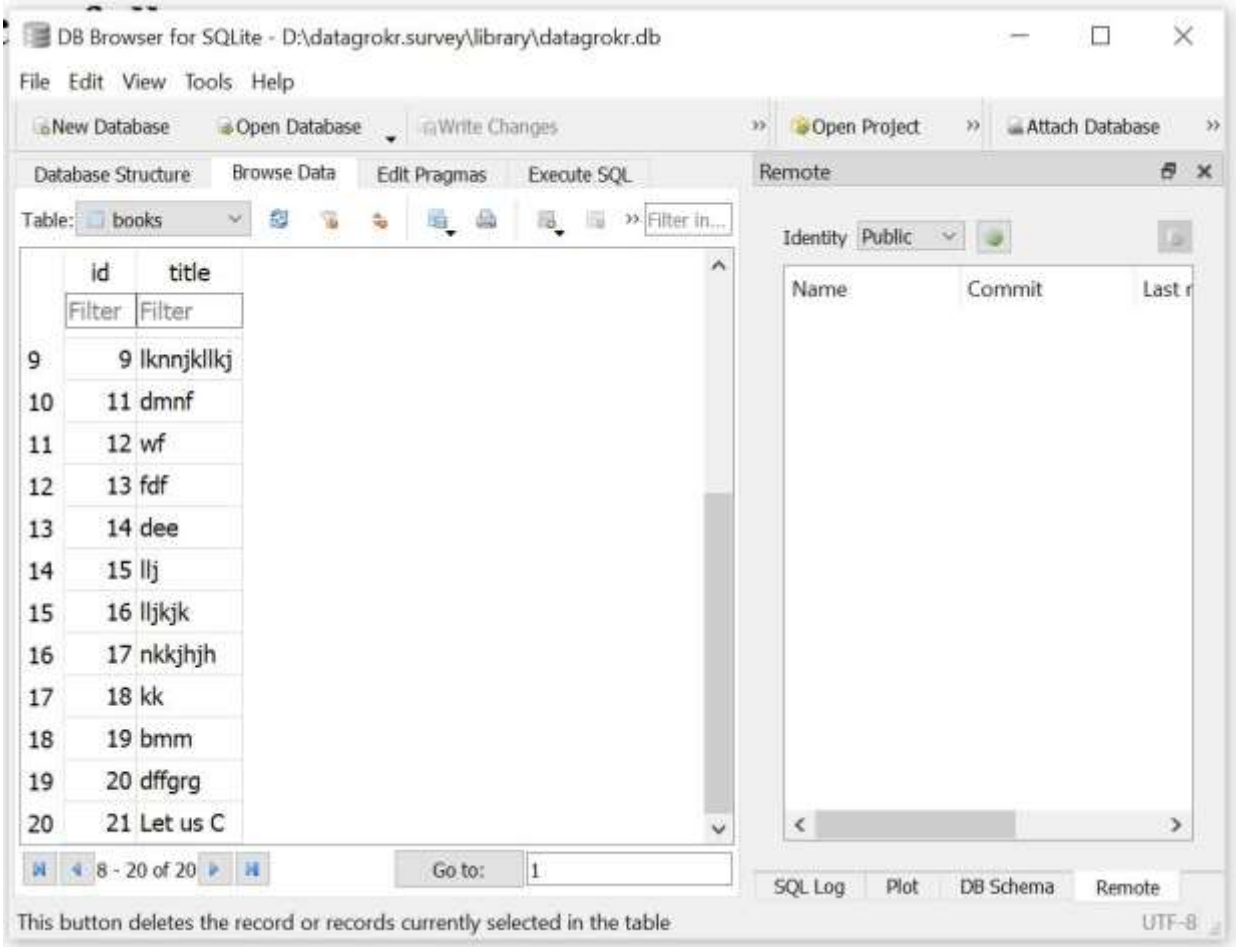
127.0.0.1:5000/updateinfo

127.0.0.1:5000/updateinfo

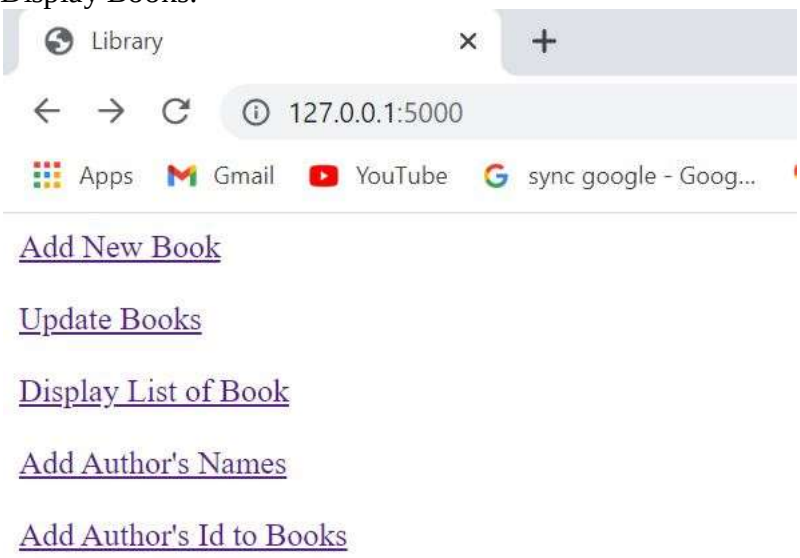
Apps Gmail YouTube sync google - Goog... Maps Infor

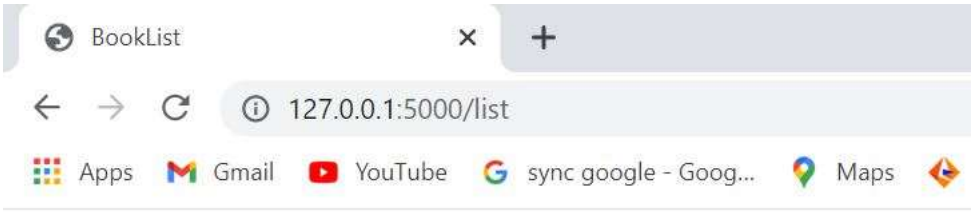
Record updated successfully

Go to home



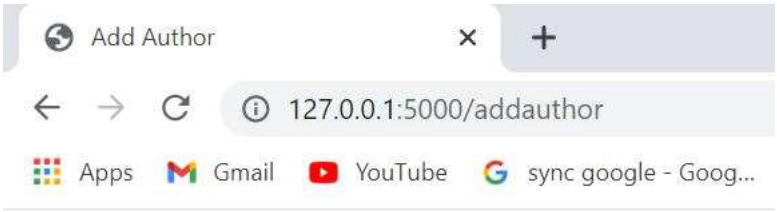
Display Books:





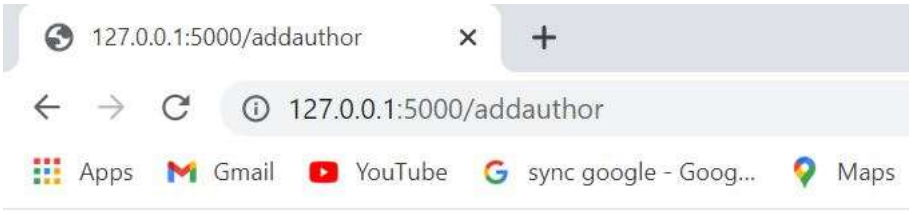
Id	Book name
1	hjh
2	test
3	nm
5	test
6	ghhh
4	kk
7	mmn
8	nkjkljkljkl
9	lknnjkllkj
11	dmnf
12	wf
13	fdf
14	dee
15	llj
16	lljkjk
17	nkkjhjh
18	kk
19	bmm
20	dffgrg
21	Let us C

Go to Home



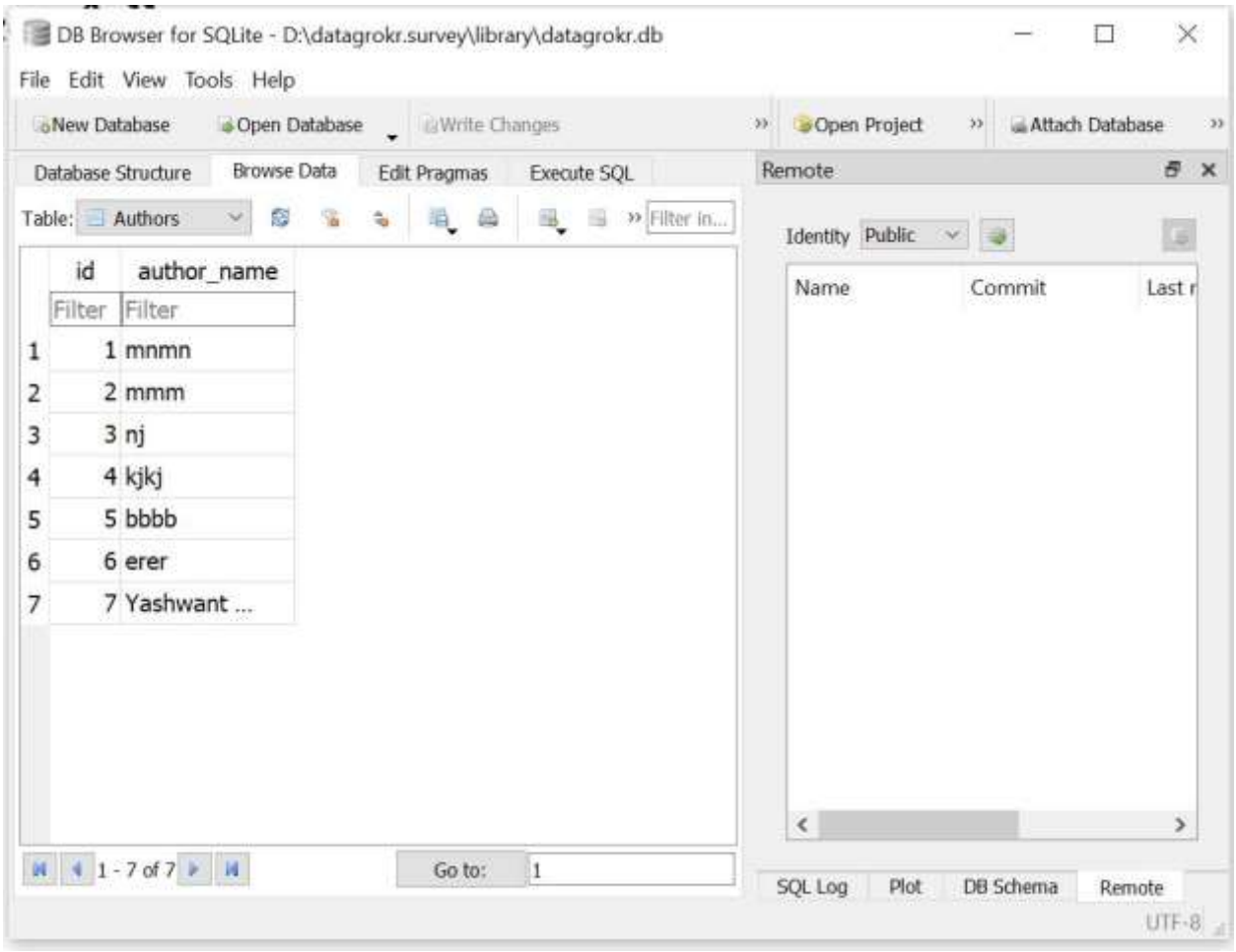
Add Author's Names

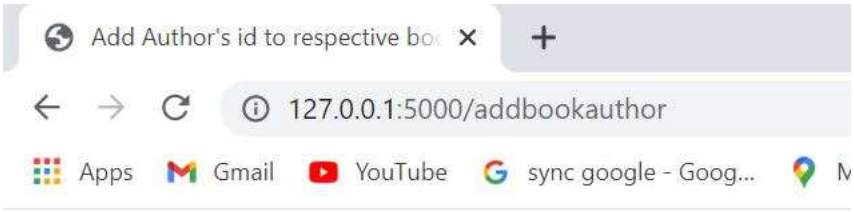
Author_Id
7
Names
Yashwant Kanetkar
submit



Record inserted successfully

Go to home


DB Browser for SQLite - D:\datagrokr.survey\library\datagrokr.db
File Edit View Tools Help
New Database
Open Database
Write Changes
Open Project
Attach Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: Authors
Filter in...
id
author_name
Filter
Filter
1 1 mnmn
2 2 mmm
3 3 nj
4 4 kjkj
5 5 bbbb
6 6 erer
7 7 Yashwant ...
1 - 7 of 7
Go to: 1
Remote
Identity Public
Name
Commit
Last r
SQL Log
Plot
DB Schema
Remote
UTF-8

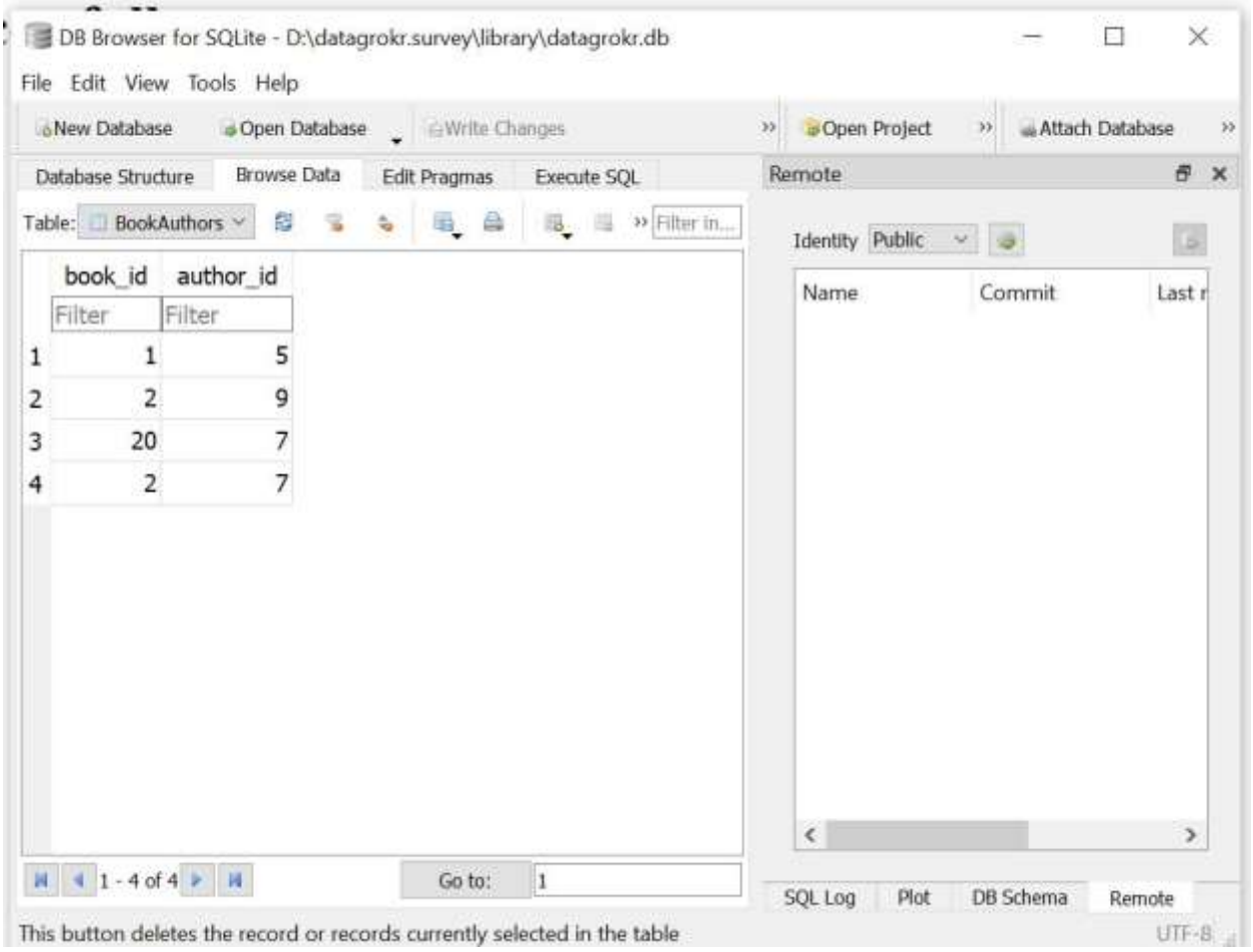


Add Author's id

Book_Id
2
Author_id
7
submit

Record inserted successfully

Go to home


DB Browser for SQLite - D:\datagrokr.survey\library\datagrokr.db
File Edit View Tools Help
New Database
Open Database
Write Changes
Open Project
Attach Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: BookAuthors
Filter in...
book_id
author_id
Filter
Filter
1 1 5
2 2 9
3 20 7
4 2 7
1 - 4 of 4
Go to: 1
Remote
Identity Public
Name
Commit
Last
SQL Log
Plot
DB Schema
Remote
This button deletes the record or records currently selected in the table
UTF-8