

Object Oriented Programming (Using Python)

UNIT- V

Using Databases with Python:

- Using Databases
- Single Table CRUD (Create, Read, Update, and Delete)
- Designing and representing a data model
- Reconstructing data with JOIN
- Many-to-many relationships

Prof. R. MADANA MOHANA

Professor, Artificial Intelligence & Data Science

<http://rmadanamohana.com/>

References

<https://www.coursera.org/learn/python-databases/home/info>

<https://www.py4e.com/book>

http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf

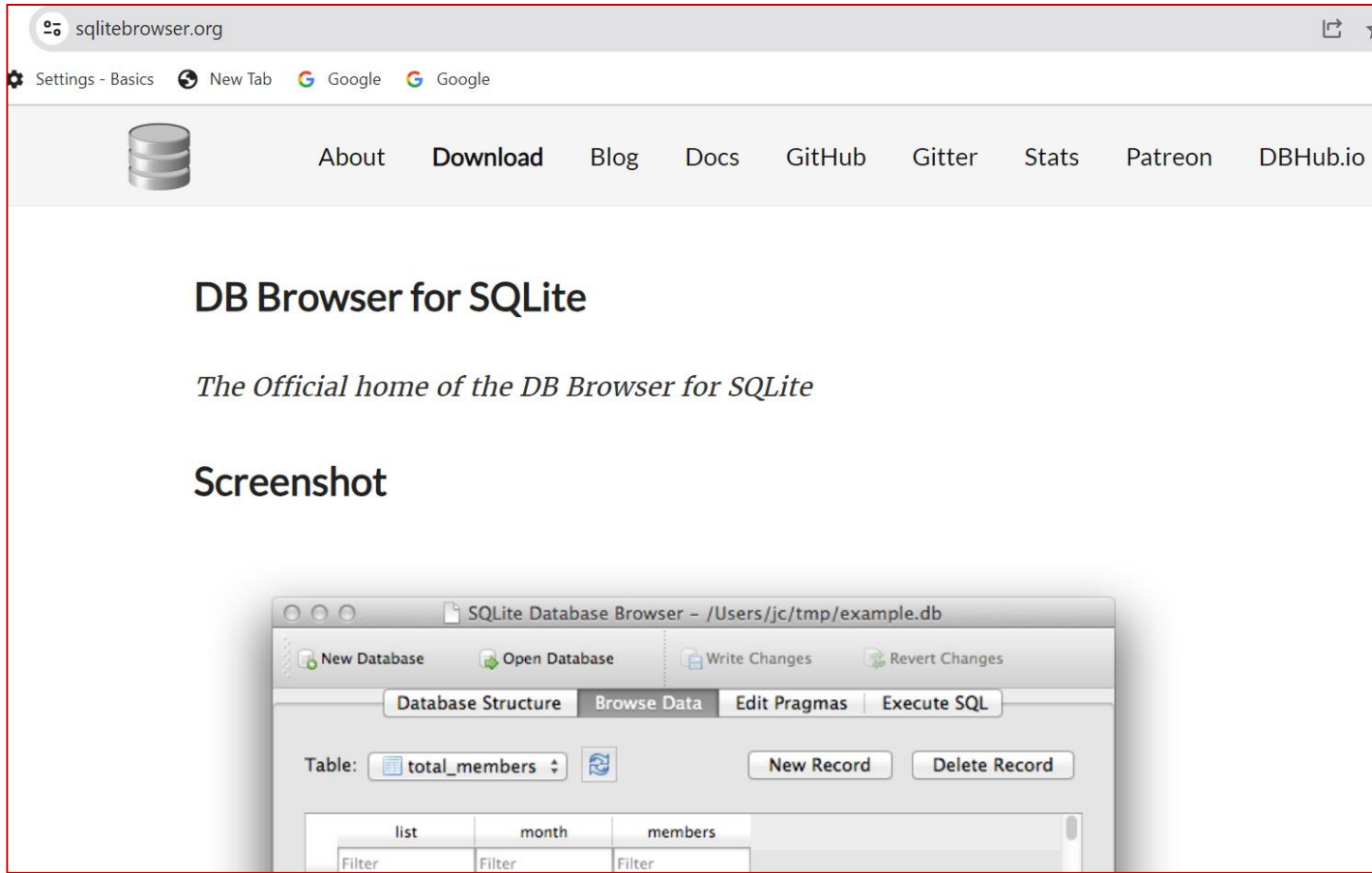
<https://www.py4e.com/code3/>

<https://www.py4e.com/lectures3/>

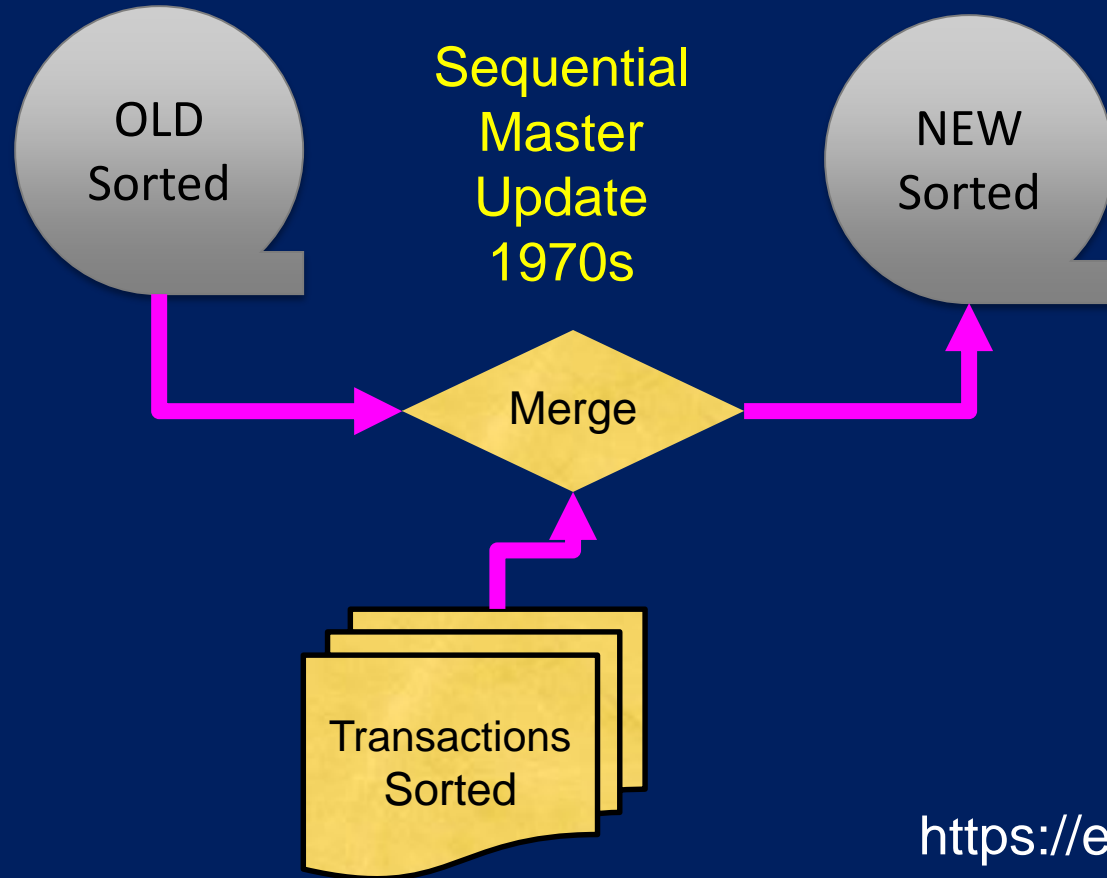
<https://www.youtube.com/playlist?list=PLIRFEj9H3Oj7Bp8-DfGpfAfDBibIRfl5p>

<http://sqlitebrowser.org/>

Relational Databases and SQLite



<http://sqlitebrowser.org/>



https://en.wikipedia.org/wiki/IBM_729

Random Access

- When you can randomly access data...
- How can you layout data to be most efficient?
- Sorting might not be the best idea



https://en.wikipedia.org/wiki/Hard_disk_drive_platter

Relational Databases

Relational databases model data by storing rows and columns in tables. The power of the relational database lies in its ability to efficiently retrieve data from those tables and in particular where there are multiple tables and the relationships between those tables involved in the query.

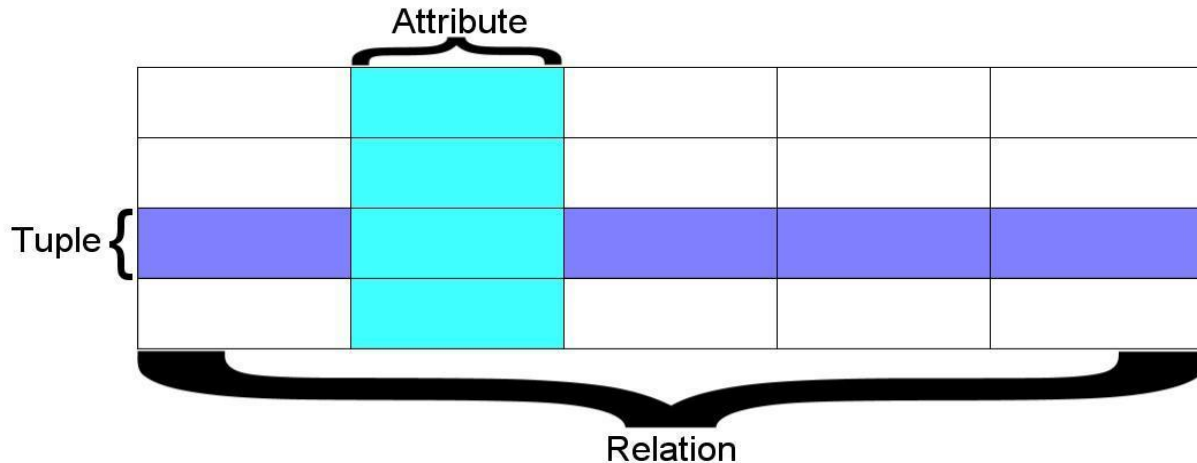
http://en.wikipedia.org/wiki/Relational_database

Terminology

- **Database** - contains many tables
- **Relation (or table)** - contains **tuples** and **attributes**
- **Tuple (or row)** - a set of **fields** that generally represents an “**object**” like a person or a music track
- **Attribute (also column or field)** - one of possibly many elements of data corresponding to the object represented by the **row**

Relation

A **relation** is defined as a set of **tuples** that have the same **attributes**. A **tuple** usually represents an **object** and information about that object. **Objects** are typically physical objects or concepts. A **relation** is usually described as a **table**, which is organized into **rows** and **columns**. All the data referenced by an **attribute** are in the same domain and **conform to the same constraints**.
(Wikipedia)



Relation - Example

The screenshot shows a Microsoft Excel spreadsheet titled "SI502 - Database". The spreadsheet contains a table with the following data:

TITLE	RATING	LEN
About to Rock	3	354
Who Made Who	4	252

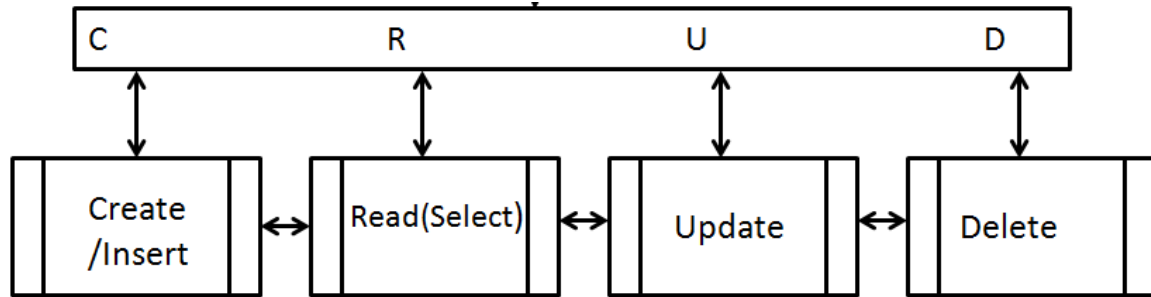
Red boxes highlight the following concepts:

- Columns / Attributes**: A box around the header row (TITLE, RATING, LEN).
- Rows / Tuples**: A box around the data rows (About to Rock, Who Made Who).
- Tables / Relations**: A box around the entire data table.

SQL

SQL (Structured Query Language) is the language we use to issue commands to the database

- Create data (also known as Add/Insert)
- Retrieve data
- Update data
- Delete data



<http://en.wikipedia.org/wiki/SQL>

Using Databases

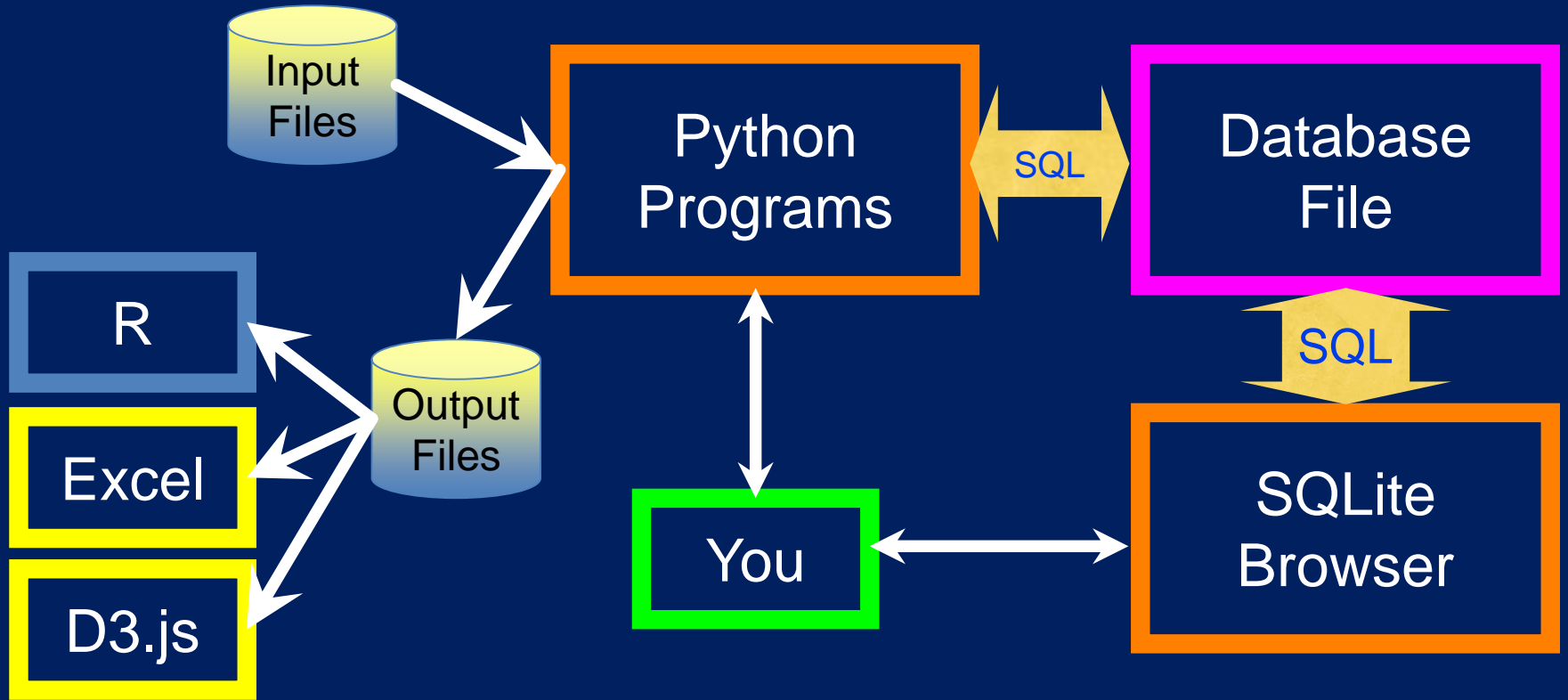
Two Roles in Large Projects

- **Application Developer** - Builds the logic for the application, the look and feel of the application - monitors the application for problems
- **Database Administrator** - Monitors and adjusts the database as the program runs in production
- Often both people participate in the building of the “**Data model**”

Database Administrator (DBA)

- A **Database Administrator (DBA)** is a person responsible for the design, implementation, maintenance, and repair of an organization's database.
- The **role** includes the **development** and **design** of **database strategies**, **monitoring** and **improving database performance** and **capacity**, and **planning** for future expansion requirements.
- They may also **plan**, **coordinate**, and **implement** **security measures** to **safeguard** the database.

Data Analysis Structure



Database Model

- A **database model** or **database schema** is the **structure** or **format of a database**, described in a formal language supported by the **database management system**.
- In other words, a “**database model**” is the application of a data model when used in conjunction with a **database management system**.

Common Database Systems

- Three major **Database Management Systems** in wide use
 - ✓ **Oracle** - Large, commercial, enterprise-scale, very very tweakable
 - ✓ **MySQL** - Simpler but very fast and scalable - commercial open source
 - ✓ **SqlServer** - Very nice - from Microsoft (also Access)
- Many other smaller projects, free and open source
 - ✓ **HSQL (Hyper SQL)**, **SQLite**, **PostgreSQL (or Postgres)**- is a powerful, open source **object-relational database system** etc.

SQLite is in Lots of Software...



symbian



python™



skype™



Microsoft®



McAfee®



Adobe®



php



Google™



TOSHIBA



Sun
microsystems

<http://www.sqlite.org/famous.html>

SQLite Browser

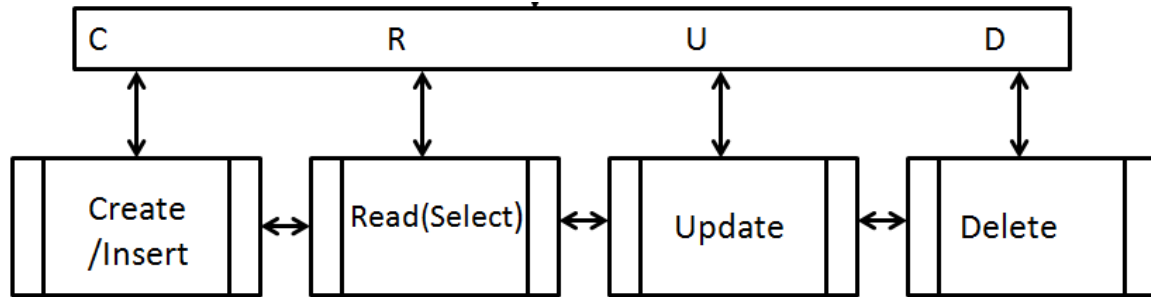
- **SQLite** is a very popular database - it is free and fast and small
- **SQLite** Browser allows us to directly manipulate **SQLite** files
<http://sqlitebrowser.org/>
- **SQLite** is embedded in **Python** and a number of other languages

Single Table CRUD (Create, Read, Update, and Delete)

SQL

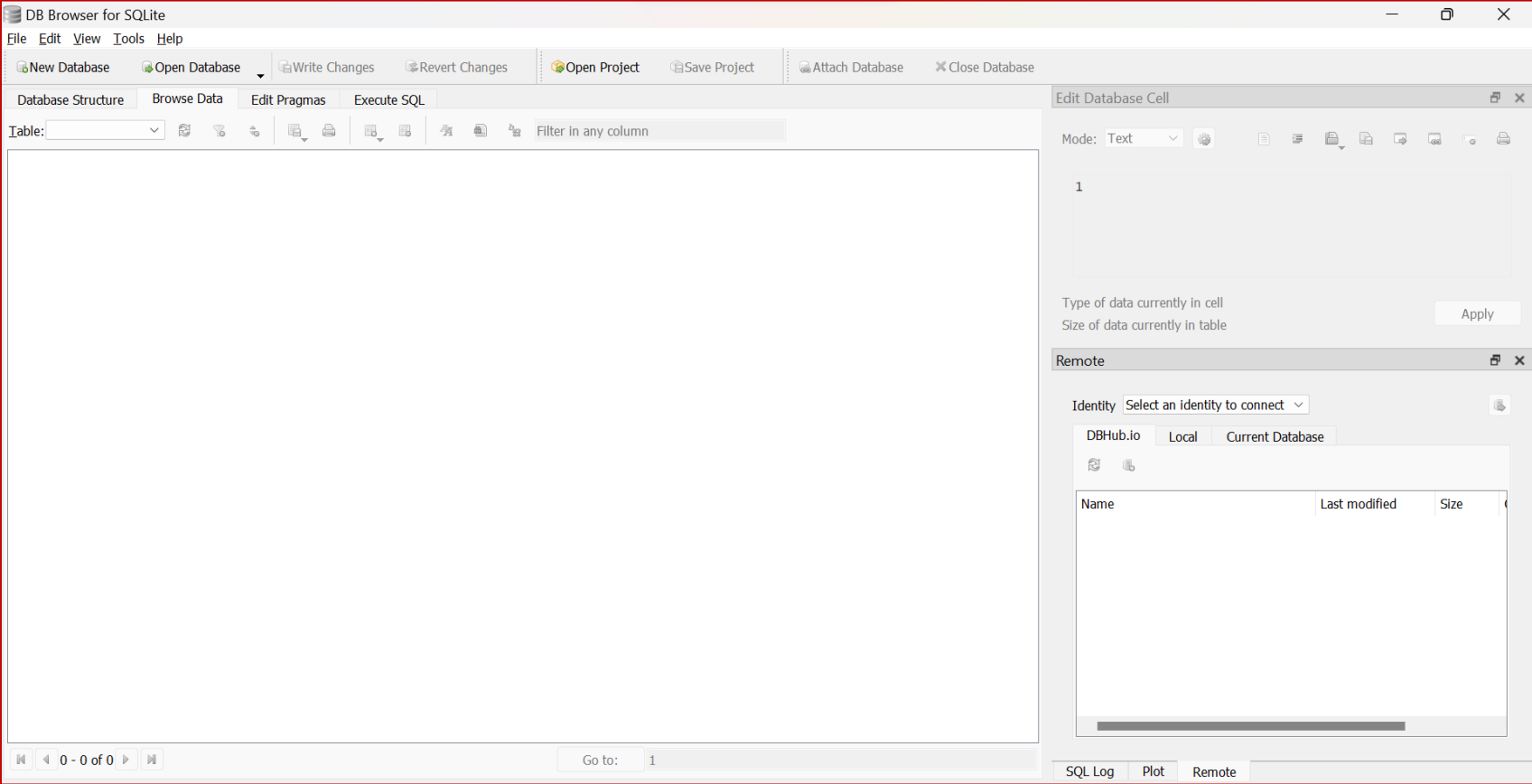
SQL (Structured Query Language) is the language we use to issue commands to the database

- Create data (also known as Add/Insert)
- Retrieve data (SELECT)
- Update data
- Delete data

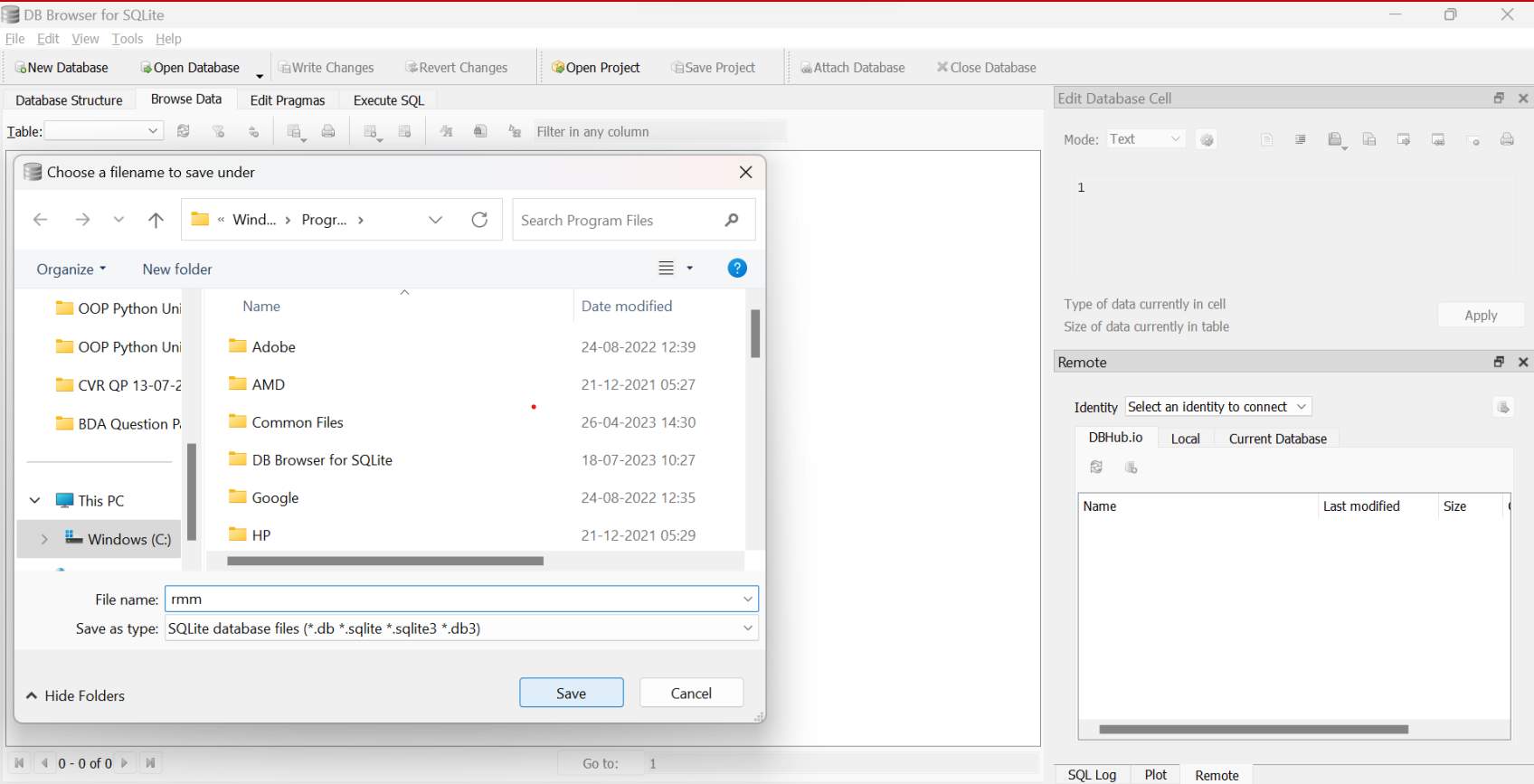


<http://en.wikipedia.org/wiki/SQL>

SQLite Browser



SQLite Browser



SQL - Create

The **CREATE** statement creates a table

```
CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
```

The screenshot shows the DB Browser for SQLite interface. The main window displays the SQL editor with the following code:

```
1 CREATE TABLE Users (  
2   name VARCHAR(128),  
3   email VARCHAR(128)  
4 )  
5
```

Below the editor, the execution results are shown:

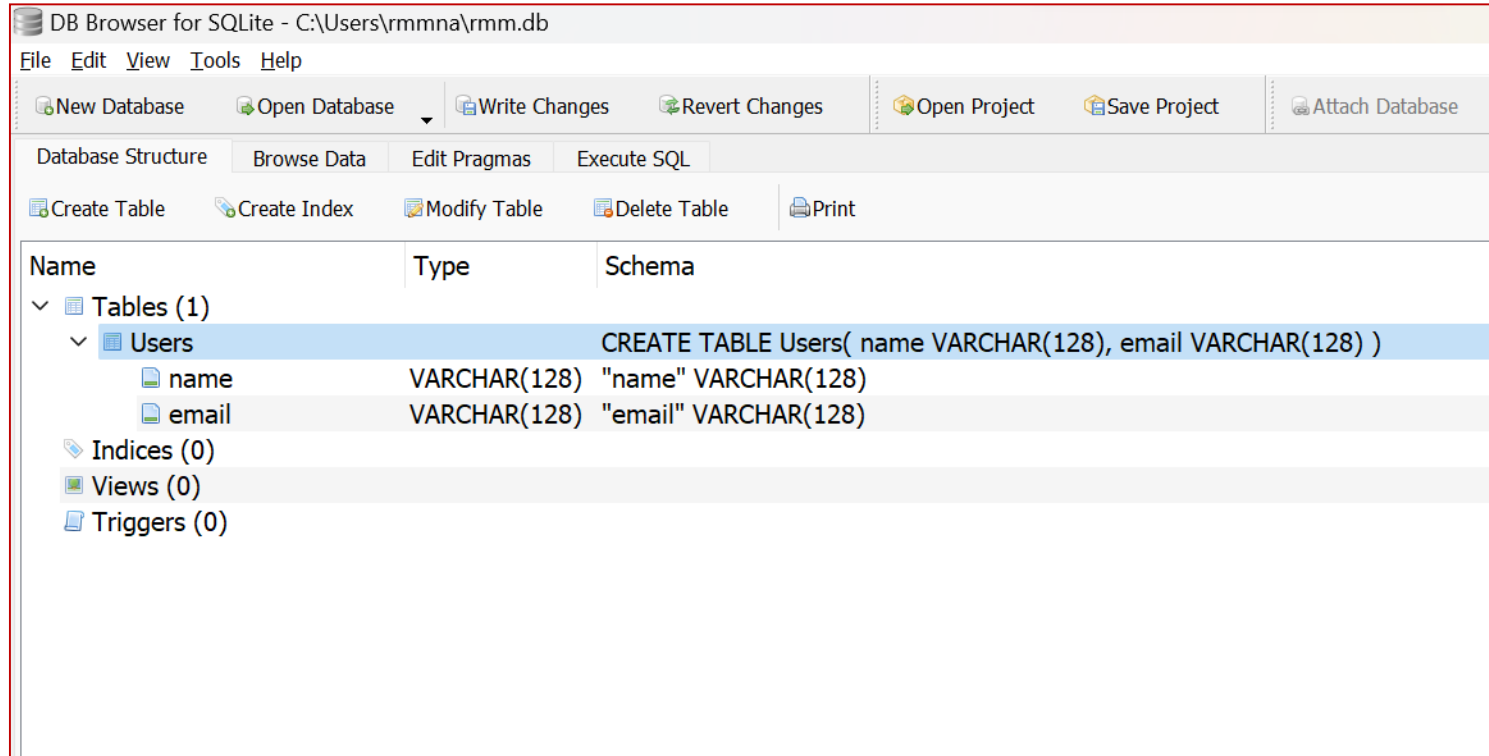
```
Execution finished without errors.  
Result: query executed successfully. Took 7ms  
At line 1:  
CREATE TABLE Users(  
  name VARCHAR(128),  
  email VARCHAR(128)  
)
```

The right-hand side of the interface shows the 'Edit Database Cell' panel, which is currently displaying 'NULL' and 'Type of data currently in cell: NULL 0 byte(s)'. Below this is the 'Remote' panel, which includes a dropdown for 'Identity' and a table with columns 'Name', 'Last modified', 'Size', and 'Com'.

SQL - Create

The **CREATE** statement creates a table

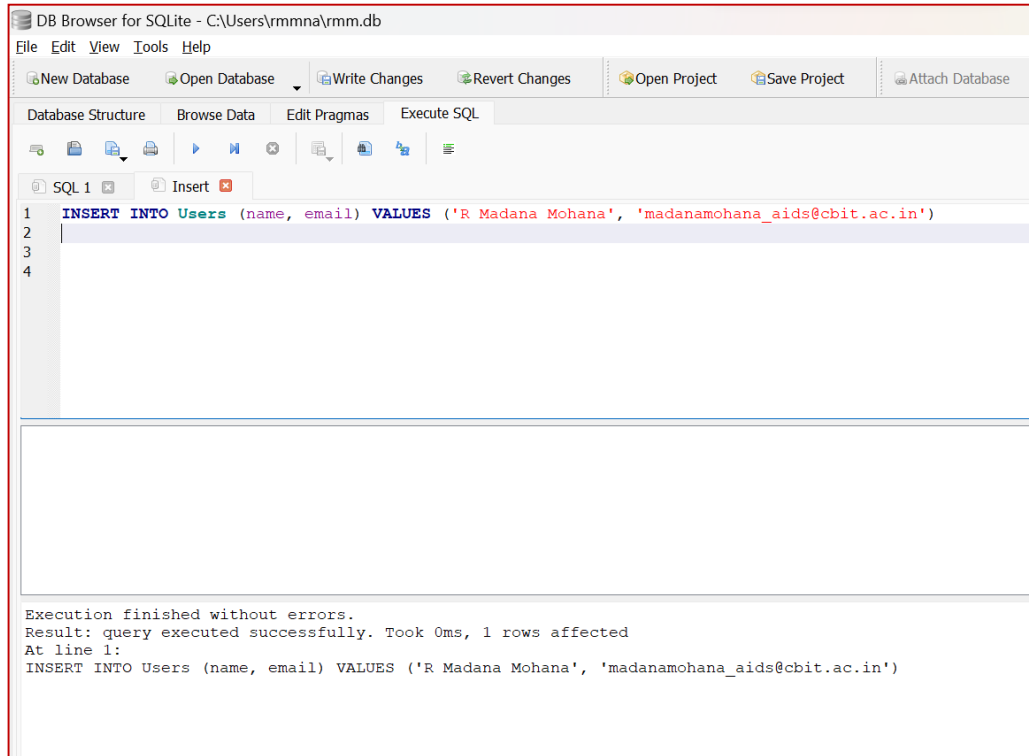
```
CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
```



SQL - Insert

The **Insert** statement **inserts** a **row** into a **table**

```
INSERT INTO Users (name, email) VALUES ('R Madana Mohana',  
'madanamohana_aid@cbit.ac.in')
```



The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database path: C:\Users\rmmna\rmm.db. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains buttons for New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, and Attach Database. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, showing a single SQL statement in a text area: `INSERT INTO Users (name, email) VALUES ('R Madana Mohana', 'madanamohana_aid@cbit.ac.in')`. Below the text area, there is a large empty space, likely for the query results. At the bottom of the window, a status bar displays the execution output: `Execution finished without errors. Result: query executed successfully. Took 0ms, 1 rows affected At line 1: INSERT INTO Users (name, email) VALUES ('R Madana Mohana', 'madanamohana_aid@cbit.ac.in')`

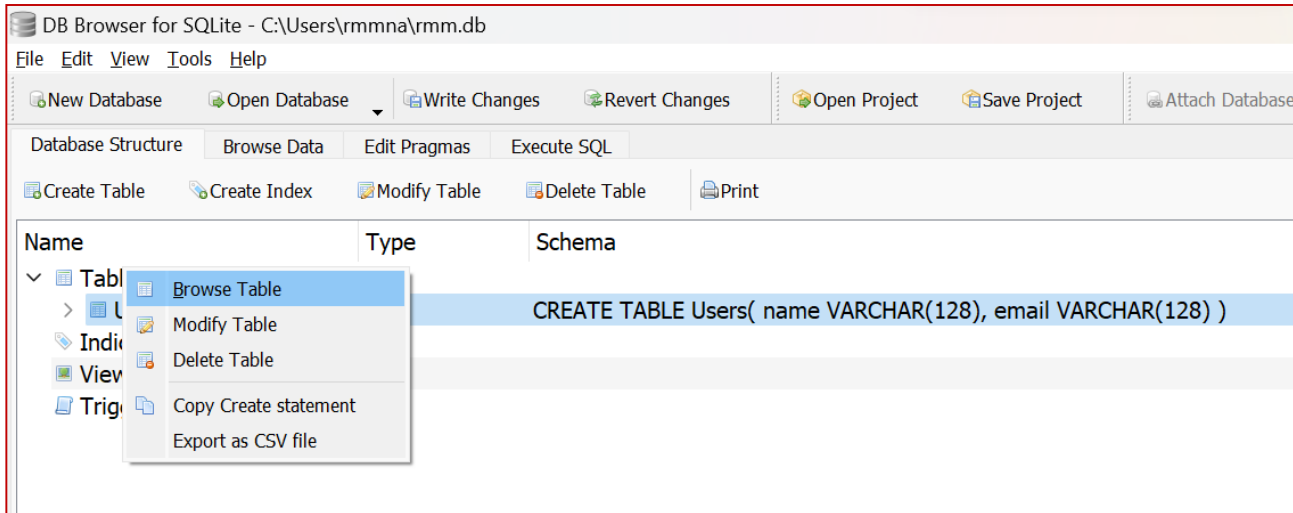
SQL - Insert

The **Insert** statement **inserts** a **row** into a **table**

```
INSERT INTO Users (name, email) VALUES ('R Madana Mohana',  
'madanamohana_aids@cbit.ac.in')
```

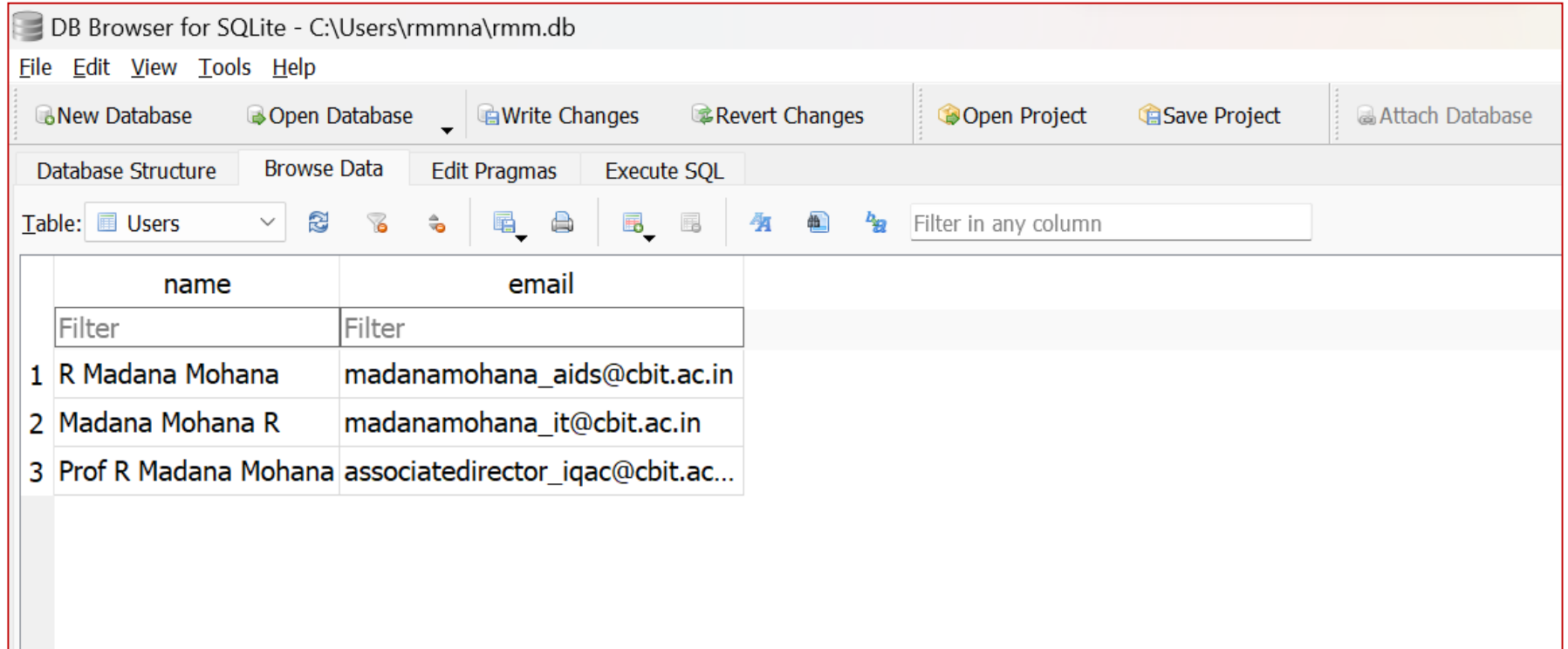
```
INSERT INTO Users (name, email) VALUES ('Madana Mohana R',  
'madanamohana_it@cbit.ac.in')
```

```
INSERT INTO Users (name, email) VALUES ('Prof R Madana Mohana',  
'associatedirector_igac@cbit.ac.in')
```



SQL - Insert

The **Insert** statement **inserts** a **row** into a **table**



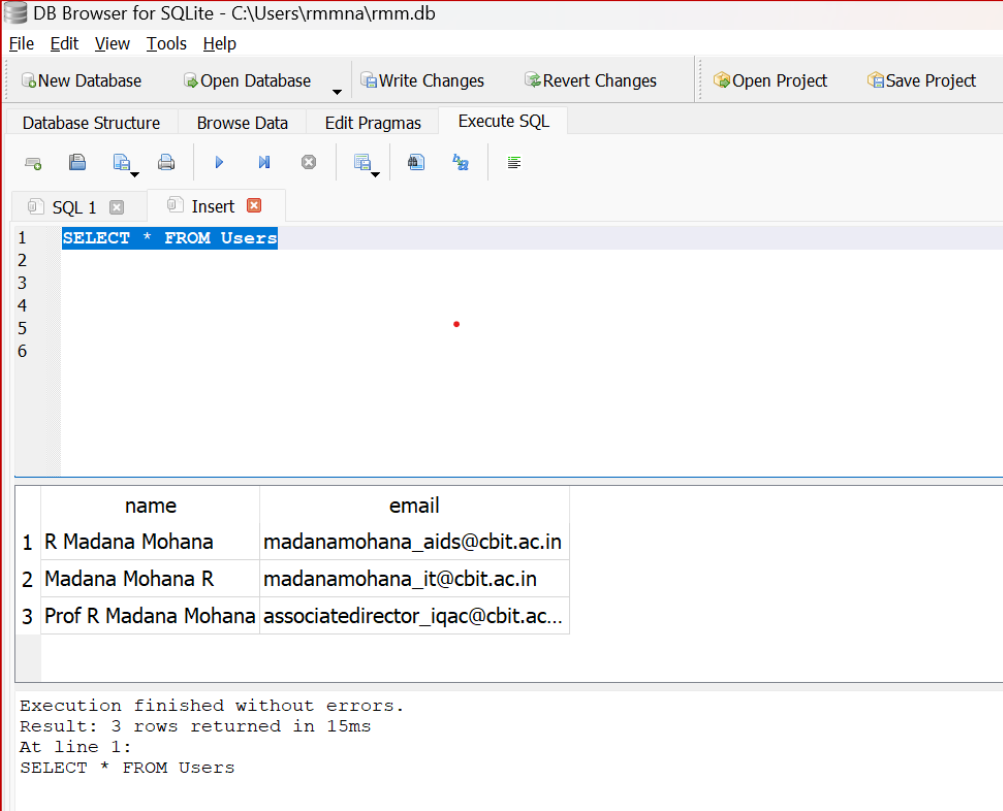
The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database is 'C:\Users\rmmna\rmm.db'. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains buttons for New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, and Attach Database. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The 'Table:' dropdown is set to 'Users'. Below the table, there are icons for refresh, zoom, and other actions, along with a search filter input field labeled 'Filter in any column'. The table data is as follows:

	name	email
	Filter	Filter
1	R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Madana Mohana R	madanamohana_it@cbit.ac.in
3	Prof R Madana Mohana	associatedirector_iqac@cbit.ac...

SQL – Retrieving Records: Select

The **SELECT** statement retrieves a group of records - we can either **retrieve all** the records or a **subset of the records** with a **WHERE** clause.

```
SELECT * FROM Users
```



DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 x Insert x

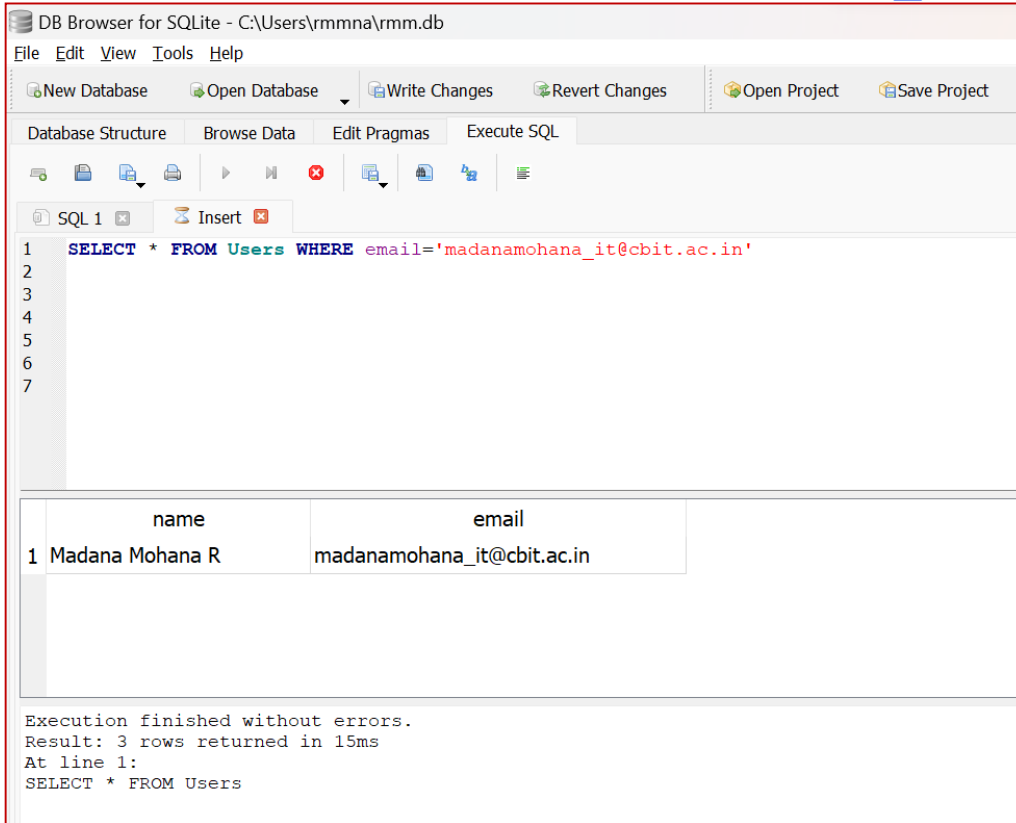
```
1 SELECT * FROM Users
2
3
4
5
6
```

	name	email
1	R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Madana Mohana R	madanamohana_it@cbit.ac.in
3	Prof R Madana Mohana	associatedirector_iqac@cbit.ac...

Execution finished without errors.
Result: 3 rows returned in 15ms
At line 1:
SELECT * FROM Users

SQL – Retrieving Records: Select

```
SELECT * FROM Users WHERE email='madanamohana_it@cbit.ac.in'
```



DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragas Execute SQL

SQL 1 Insert

```
1 SELECT * FROM Users WHERE email='madanamohana_it@cbit.ac.in'
```

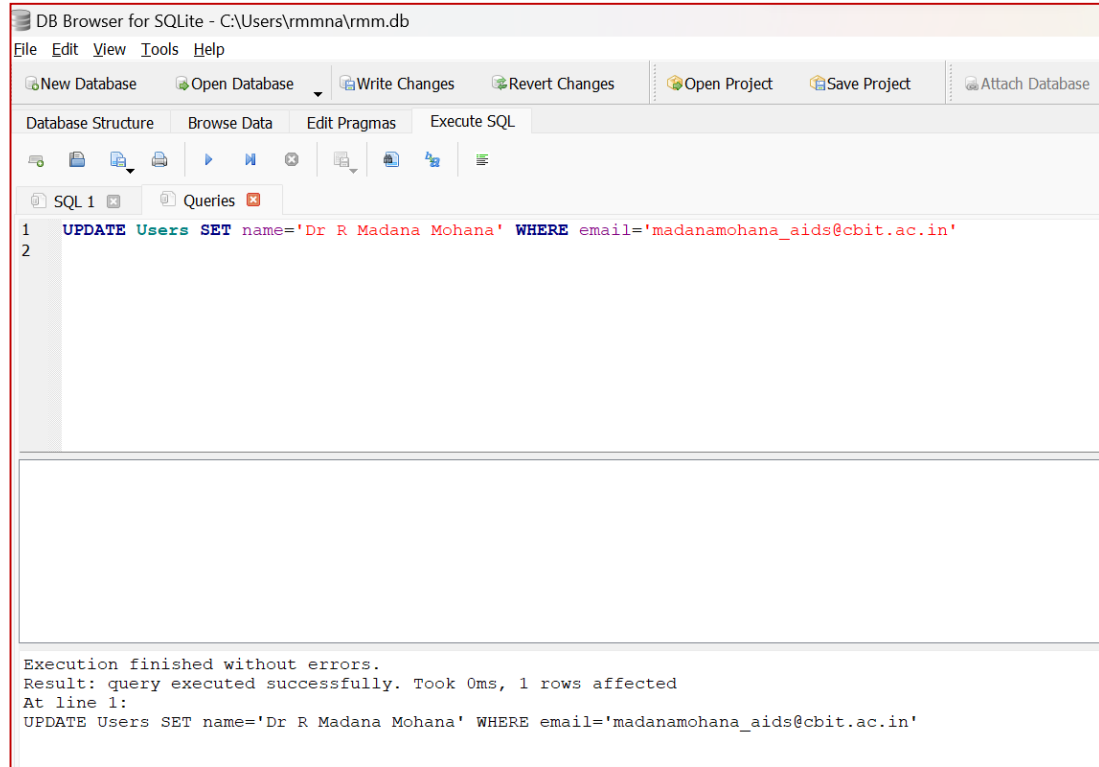
	name	email
1	Madana Mohana R	madanamohana_it@cbit.ac.in

Execution finished without errors.
Result: 3 rows returned in 15ms
At line 1:
SELECT * FROM Users

SQL – Update

Allows the **updating** of a field with a **WHERE** clause

```
UPDATE Users SET name='Dr R Madana Mohana' WHERE  
email='madanamohana_aids@cbit.ac.in'
```



The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database path: C:\Users\ymmna\ymmna.db. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains buttons for New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, and Attach Database. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The SQL editor shows the following query:

```
1 UPDATE Users SET name='Dr R Madana Mohana' WHERE email='madanamohana_aids@cbit.ac.in'  
2
```

Below the editor, the execution results are displayed:

```
Execution finished without errors.  
Result: query executed successfully. Took 0ms, 1 rows affected  
At line 1:  
UPDATE Users SET name='Dr R Madana Mohana' WHERE email='madanamohana_aids@cbit.ac.in'
```

SQL – Update

Output before and after Update:

```
select * from Users
```

DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 Insert

```
1 SELECT * FROM Users
```

	name	email
1	R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Madana Mohana R	madanamohana_it@cbit.ac.in
3	Prof R Madana Mohana	associatedirector_iqac@cbit.ac...

Execution finished without errors.
Result: 3 rows returned in 15ms
At line 1:
SELECT * FROM Users

DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 Queries

```
1 select * from Users
```

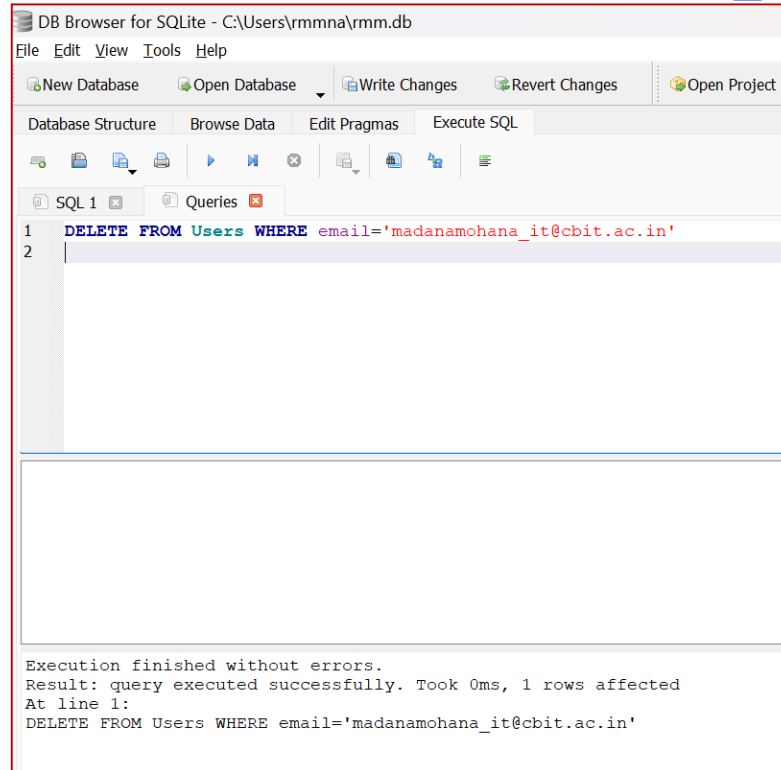
	name	email
1	Dr R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Madana Mohana R	madanamohana_it@cbit.ac.in
3	Prof R Madana Mohana	associatedirector_iqac@cbit.ac.in

Execution finished without errors.
Result: 3 rows returned in 10ms
At line 1:
select * from Users

SQL – Delete

Deletes a row in a table based on selection criteria

```
DELETE FROM Users WHERE email='madanamohana_it@cbit.ac.in'
```



The screenshot shows the 'DB Browser for SQLite' application window. The title bar indicates the database file is 'C:\Users\rmmna\rmm.db'. The menu bar includes 'File', 'Edit', 'View', 'Tools', and 'Help'. Below the menu bar are buttons for 'New Database', 'Open Database', 'Write Changes', 'Revert Changes', and 'Open Project'. The main interface has tabs for 'Database Structure', 'Browse Data', 'Edit Pragmas', and 'Execute SQL'. The 'Execute SQL' tab is active, showing a text editor with the following SQL query:

```
1 DELETE FROM Users WHERE email='madanamohana_it@cbit.ac.in'
```

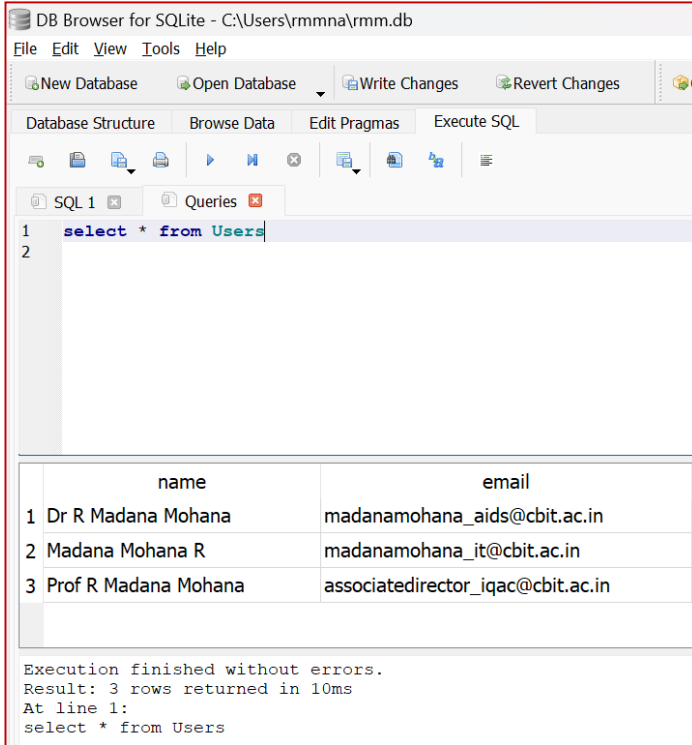
Below the editor, the execution results are displayed:

```
Execution finished without errors.  
Result: query executed successfully. Took 0ms, 1 rows affected  
At line 1:  
DELETE FROM Users WHERE email='madanamohana_it@cbit.ac.in'
```

SQL – Delete

Output Before and after Delete

```
select * from Users
```



DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

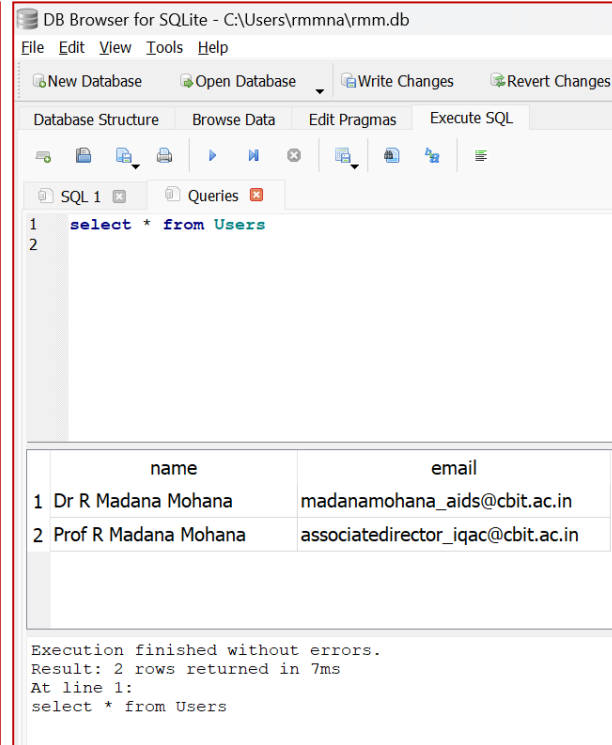
New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

```
SQL 1
1 select * from Users
2
```

	name	email
1	Dr R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Madana Mohana R	madanamohana_it@cbit.ac.in
3	Prof R Madana Mohana	associatedirector_iqac@cbit.ac.in

Execution finished without errors.
Result: 3 rows returned in 10ms
At line 1:
select * from Users



DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

```
SQL 1
1 select * from Users
2
```

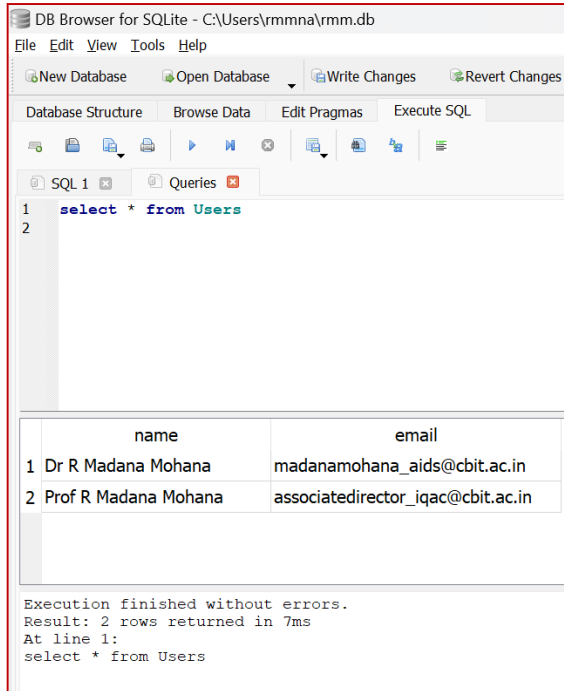
	name	email
1	Dr R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Prof R Madana Mohana	associatedirector_iqac@cbit.ac.in

Execution finished without errors.
Result: 2 rows returned in 7ms
At line 1:
select * from Users

Sorting with ORDER BY

We can add an **ORDER BY** clause to **SELECT** statements to get the results **sorted** in **ascending** or **descending** order.

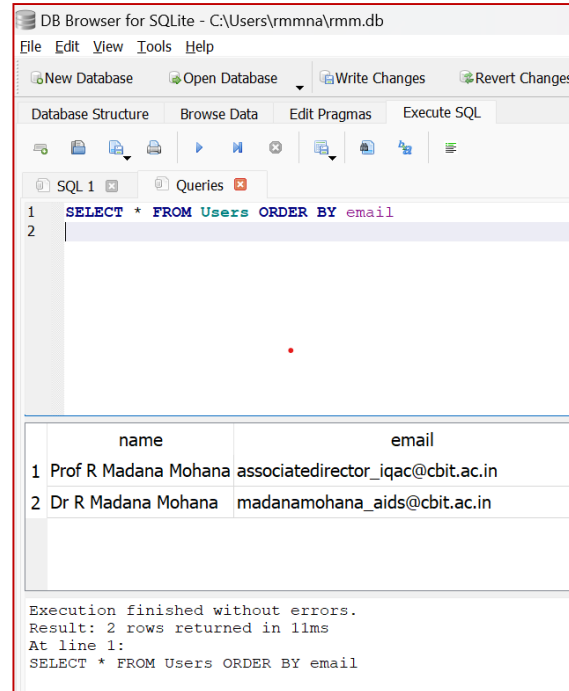
```
SELECT * FROM Users ORDER BY email
```



The screenshot shows the DB Browser for SQLite interface. The SQL editor contains the query: `select * from Users`. The results table below shows two rows of data:

	name	email
1	Dr R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Prof R Madana Mohana	associatedirector_iqac@cbit.ac.in

Execution finished without errors.
Result: 2 rows returned in 7ms
At line 1:
select * from Users



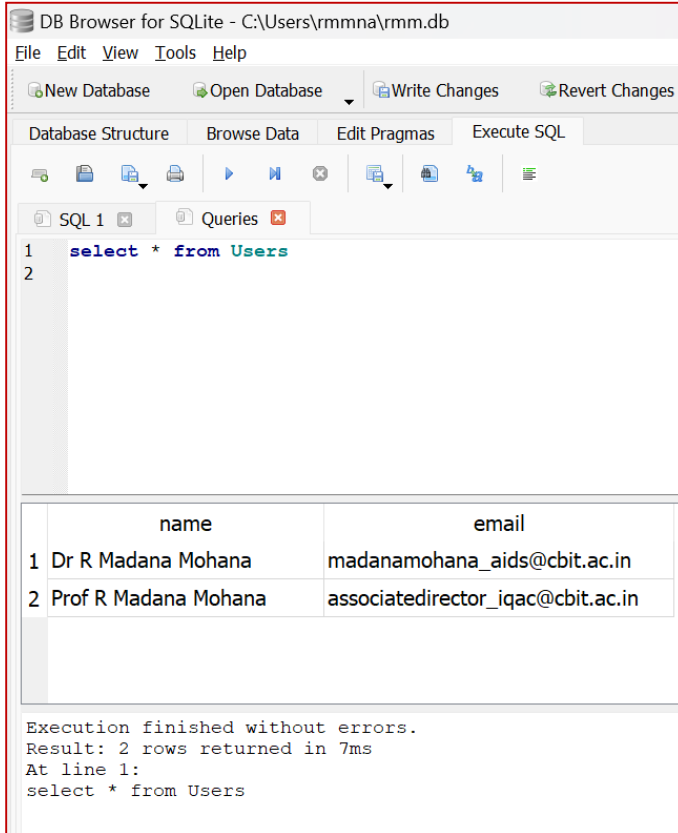
The screenshot shows the DB Browser for SQLite interface. The SQL editor contains the query: `SELECT * FROM Users ORDER BY email`. The results table below shows two rows of data, sorted by email:

	name	email
1	Prof R Madana Mohana	associatedirector_iqac@cbit.ac.in
2	Dr R Madana Mohana	madanamohana_aids@cbit.ac.in

Execution finished without errors.
Result: 2 rows returned in 11ms
At line 1:
SELECT * FROM Users ORDER BY email

Sorting with ORDER BY

SELECT * FROM Users ORDER BY name DESC



DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

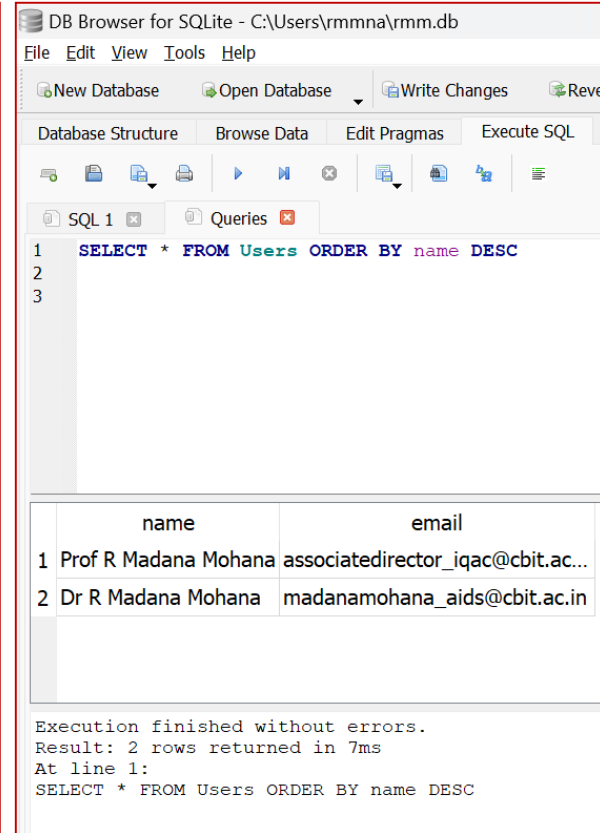
Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 Queries

```
1 select * from Users
2
```

	name	email
1	Dr R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Prof R Madana Mohana	associatedirector_iqac@cbit.ac.in

Execution finished without errors.
Result: 2 rows returned in 7ms
At line 1:
select * from Users



DB Browser for SQLite - C:\Users\rmmna\rmm.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 Queries

```
1 SELECT * FROM Users ORDER BY name DESC
2
3
```

	name	email
1	Prof R Madana Mohana	associatedirector_iqac@cbit.ac...
2	Dr R Madana Mohana	madanamohana_aids@cbit.ac.in

Execution finished without errors.
Result: 2 rows returned in 7ms
At line 1:
SELECT * FROM Users ORDER BY name DESC

SQL Summary

Output Before and after Delete

```
CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
```

```
INSERT INTO Users (name, email) VALUES ('R Madana Mohana',  
'madanamohana_aids@cbit.ac.in')
```

```
SELECT * FROM Users
```

```
SELECT * FROM Users WHERE email='madanamohana_it@cbit.ac.in'
```

```
UPDATE Users SET name='Dr R Madana Mohana' WHERE  
email='madanamohana_aids@cbit.ac.in'
```

```
DELETE FROM Users WHERE email='madanamohana_it@cbit.ac.in'
```

```
SELECT * FROM Users ORDER BY email
```

```
SELECT * FROM Users ORDER BY name DESC
```

Object Oriented Programming (Using Python)

Single Table CRUD
(Create, Read, Update, and Delete)

Python Demo-1

Single Table CRUD – Python Demo

CREATE: `sqlite_create.py`

sqlite_create.py - C:\Users\rmmna\AppData\Local\Programs\Python\Python310\sqlite_create.py (3.10.0)

File Edit Format Run Options Window Help

```
import sqlite3
conn = sqlite3.connect('oop.db')
print('Opened database successfully');
conn.execute('''CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
''')
print('Table created successfully')
conn.close()
```

DB Browser for SQLite - C:\Users\rmmna\AppData\Local\Programs\Python\Python310\oop.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragma Execute SQL

Create Table Create Index Modify Table Delete Table Print

Name	Type	Schema
Tables (2)		
Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
name	VARCHAR(128)	"name" VARCHAR(128)
email	VARCHAR(128)	"email" VARCHAR(128)

Single Table CRUD – Python Demo

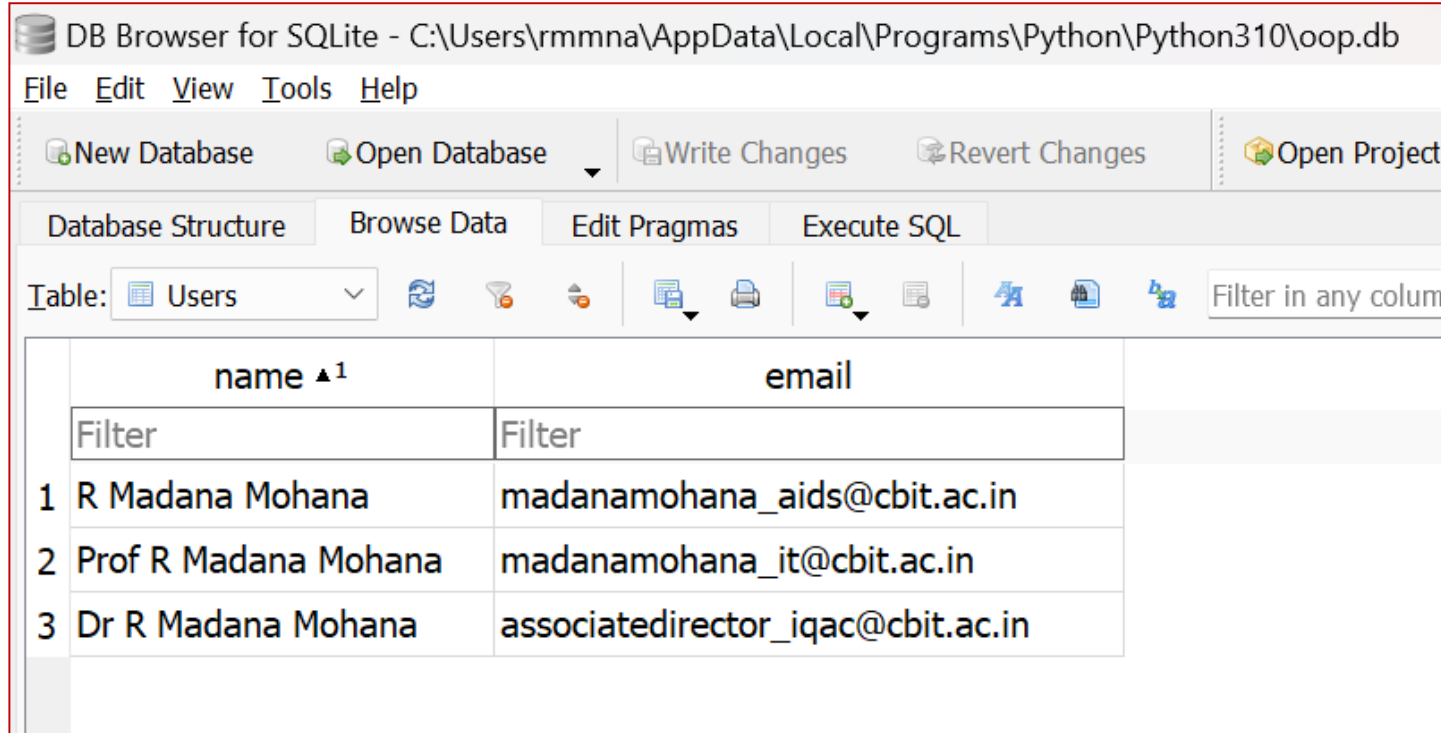
INSERT: `sqlite_insert.py`

```
sqlite_insert.py - C:\Users\rmmna\AppData\Local\Programs\Python\Python310\sqlite_insert.py (3.10.0)
File Edit Format Run Options Window Help
import sqlite3
conn = sqlite3.connect('oop.db')
print('Opened database successfully');
conn.execute("INSERT INTO Users (name, email) VALUES ('R Madana Mohana', 'madanamohana_aids@cbit.ac.in')")
conn.execute("INSERT INTO Users (name, email) VALUES ('Prof R Madana Mohana', 'madanamohana_it@cbit.ac.in')")
conn.execute("INSERT INTO Users (name, email) VALUES ('Dr R Madana Mohana', 'associatedirector_iqac@cbit.ac.in')")
conn.commit()
print('Records Inserted successfully')
conn.close()
```

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/sqlite_insert.py
Opened database successfully
Records Inserted successfully
>>>
```


Single Table CRUD – Python Demo

INSERT: `sqlite_insert.py`



The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database file path: `C:\Users\rmmna\AppData\Local\Programs\Python\Python310\oop.db`. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains buttons for New Database, Open Database, Write Changes, Revert Changes, and Open Project. Below the toolbar are tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The 'Table:' dropdown is set to 'Users'. The main area displays a table with two columns: 'name' and 'email'. The table contains three rows of data, with the first row highlighted. A 'Filter in any column' input field is visible on the right side of the table.

	name ▲ ¹	email
	Filter	Filter
1	R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Prof R Madana Mohana	madanamohana_it@cbit.ac.in
3	Dr R Madana Mohana	associatedirector_iqac@cbit.ac.in

Single Table CRUD – Python Demo

SELECT (Retrieve): `sqlite_select.py`

```
sqlite_select.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/sqlite_select.py (3.10.0)
File Edit Format Run Options Window Help
import sqlite3
conn = sqlite3.connect('oop.db')
print('Opened database successfully');
cursor=conn.execute("SELECT * from Users")
for row in cursor:
    print("Name = ", row[0])
    print("E-mail = ", row[1], "\n")
print('Records Retrieved successfully')
conn.close()
```

Single Table CRUD – Python Demo

SELECT (Retrieve): `sqlite_select.py`

```
>>> = RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/sqlite_select.py
Opened database successfully
Name = R Madana Mohana
E-mail = madanamohana_aids@cbit.ac.in

Name = Prof R Madana Mohana
E-mail = madanamohana_it@cbit.ac.in

Name = Dr R Madana Mohana
E-mail = associatedirector iqac@cbit.ac.in

Records Retrieved successfully
>>> |
```

Single Table CRUD – Python Demo

UPDATE: `sqlite_update.py`

```
sqlite_update.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/sqlite_update.py (3.10.0)
File Edit Format Run Options Window Help
import sqlite3
conn = sqlite3.connect('oop.db')
print('Opened database successfully');
cursor=conn.execute("UPDATE Users SET name='Dr R. Madana Mohana' WHERE email='madanamohana_aids@cbit.ac.in'")
conn.commit()
print("Total number of rows updated:", conn.total_changes)
cursor=conn.execute("SELECT * from Users")
for row in cursor:
    print("Name = ", row[0])
    print("E-mail = ", row[1], "\n")
print('Records Updated successfully')
conn.close()
```

Single Table CRUD – Python Demo

UPDATE: `sqlite_update.py`

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/sqlite_update.py
Opened database successfully
Total number of rows updated: 1
Name = Dr R. Madana Mohana
E-mail = madanamohana_aids@cbit.ac.in

Name = Prof R Madana Mohana
E-mail = madanamohana_it@cbit.ac.in

Name = Dr R Madana Mohana
E-mail = associatedirector_iqac@cbit.ac.in

Records Updated successfully
>>> |
```

Single Table CRUD – Python Demo

UPDATE: [sqlite_update.py](#)

DB Browser for SQLite - C:\Users\rmmna\AppData\Local\Programs\Python\Python38-64\sqlite3\sqlite3.exe

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: Users

	name ▲ ¹	email
	Filter	Filter
1	R Madana Mohana	madanamohana_aids@cbit.ac.in
2	Prof R Madana Mohana	madanamohana_it@cbit.ac.in
3	Dr R Madana Mohana	associatedirector_iqac@cbit.ac.in

DB Browser for SQLite - C:\Users\rmmna\AppData\Local\Programs\Python\Python38-64\sqlite3\sqlite3.exe

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: Users

	name ▼ ¹	email
	Filter	Filter
1	Dr R Madana Mohana	associatedirector_iqac@cbit.ac.in
2	Dr R. Madana Mohana	madanamohana_aids@cbit.ac.in
3	Prof R Madana Mohana	madanamohana_it@cbit.ac.in

Single Table CRUD – Python Demo

DELETE: `sqlite_delete.py`

```
sqlite_delete.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/sqlite_delete.py (3.10.0)
File Edit Format Run Options Window Help
import sqlite3
conn = sqlite3.connect('oop.db')
print('Opened database successfully');
cursor=conn.execute("DELETE FROM Users WHERE email='madanamohana_it@cbit.ac.in'")
conn.commit()
print("Total number of rows deleted:", conn.total_changes)
cursor=conn.execute("SELECT * from Users")
for row in cursor:
    print("Name = ", row[0])
    print("E-mail = ", row[1], "\n")
print('Records Deleted successfully')
conn.close()
```

Single Table CRUD – Python Demo

DELETE: `sqlite_delete.py`

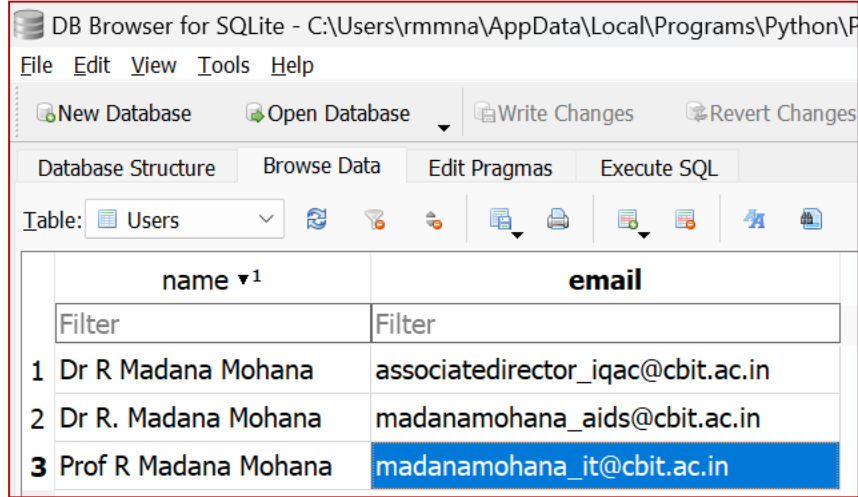
```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/sqlite_delete.py
Opened database successfully
Total number of rows deleted: 1
Name = Dr R. Madana Mohana
E-mail = madanamohana_aids@cbit.ac.in

Name = Dr R Madana Mohana
E-mail = associatedirector_iqac@cbit.ac.in

Records Deleted successfully
>>>
```


Single Table CRUD – Python Demo

DELETE: `sqlite_delete.py`



DB Browser for SQLite - C:\Users\rmmna\AppData\Local\Programs\Python\Python38\Tools\sqlitebrowser

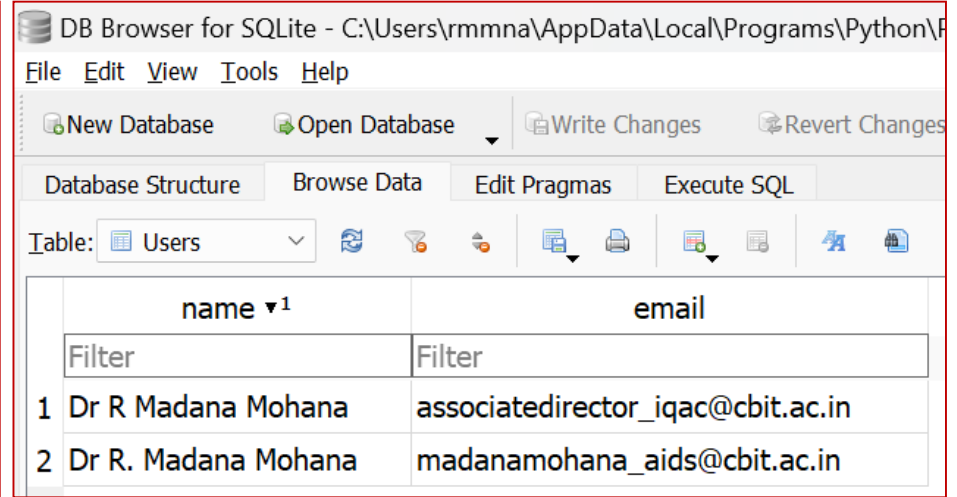
File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: Users

	name ▼ ¹	email
	Filter	Filter
1	Dr R Madana Mohana	associatedirector_iqac@cbit.ac.in
2	Dr R. Madana Mohana	madanamohana_aids@cbit.ac.in
3	Prof R Madana Mohana	madanamohana_it@cbit.ac.in



DB Browser for SQLite - C:\Users\rmmna\AppData\Local\Programs\Python\Python38\Tools\sqlitebrowser

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: Users

	name ▼ ¹	email
	Filter	Filter
1	Dr R Madana Mohana	associatedirector_iqac@cbit.ac.in
2	Dr R. Madana Mohana	madanamohana_aids@cbit.ac.in

Object Oriented Programming (Using Python)

Single Table CRUD
(Create, Read, Update, and Delete)

Python Demo-2

Single Table CRUD – Python Demo

File Name: `emaildb.py` (1/3)

emaildb.py - C:\Users\rmmna\Downloads\OOPS-PYTHON II SEM R22\OOP Python Unit-5\emaildb.py (3.10.0)

File Edit Format Run Options Window Help

```
import sqlite3
```

```
conn = sqlite3.connect('emaildb.sqlite')
```

```
cur = conn.cursor()
```

```
cur.execute('DROP TABLE IF EXISTS Counts')
```

```
cur.execute('''
```

```
CREATE TABLE Counts (email TEXT, count INTEGER)''')
```

Single Table CRUD – Python Demo

File Name: `emaildb.py` (2/3)

```
emaildb.py - C:\Users\rmmna\Downloads\OOPS-PYTHON II SEM R22\OOP Python Unit-5\emaildb.py (3.10.0)
File Edit Format Run Options Window Help

fname = input('Enter file name: ')
if (len(fname) < 1): fname = 'mbox-short.txt'
fh = open(fname)
for line in fh:
    if not line.startswith('From: '): continue
    pieces = line.split()
    email = pieces[1]
    cur.execute('SELECT count FROM Counts WHERE email = ? ', (email,))
    row = cur.fetchone()
    if row is None:
        cur.execute('''INSERT INTO Counts (email, count)
                    VALUES (?, 1)''', (email,))
    else:
        cur.execute('UPDATE Counts SET count = count + 1 WHERE email = ?'
                    (email,))
    conn.commit()
```

Single Table CRUD – Python Demo

File Name: emaildb.py (3/3)

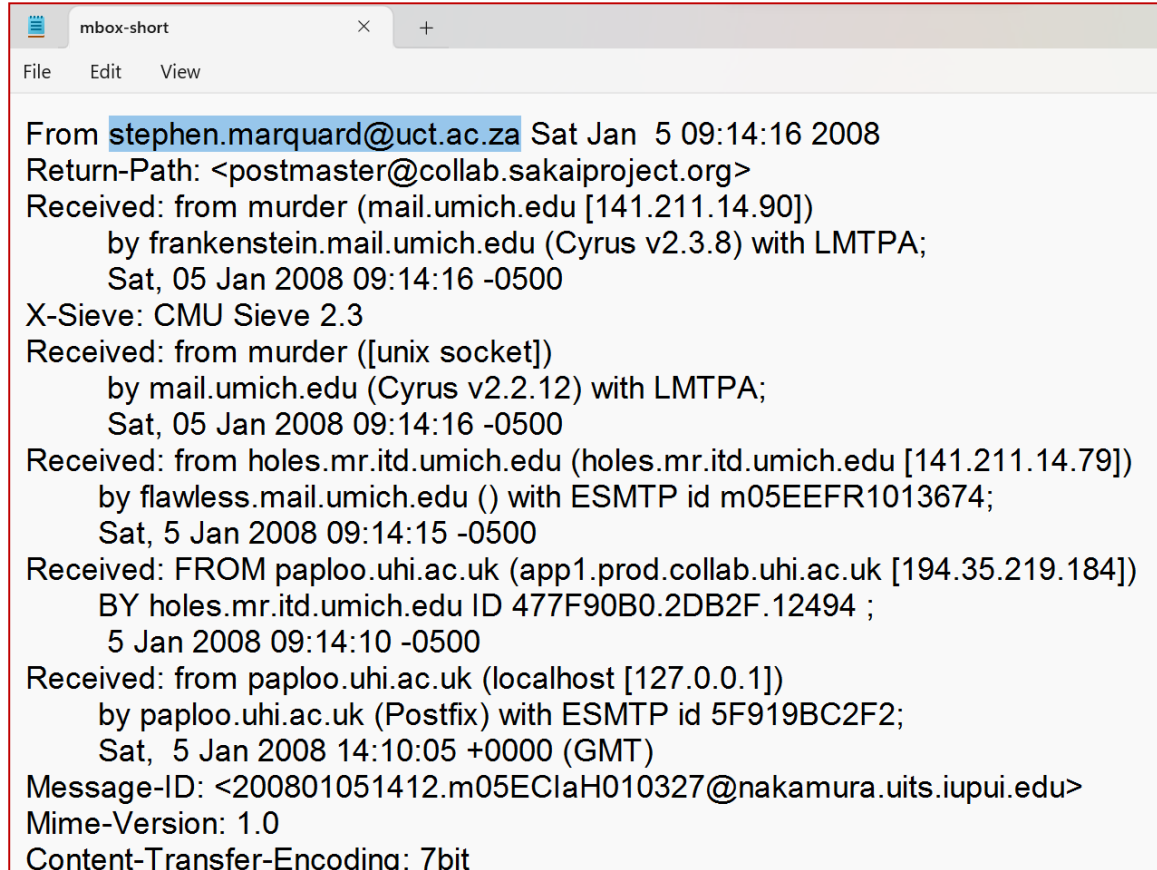
```
# https://www.sqlite.org/lang\_select.html
sqlstr = 'SELECT email, count FROM Counts ORDER BY count DESC LIMIT 10'

for row in cur.execute(sqlstr):
    print(str(row[0]), row[1])

cur.close()
```

Single Table CRUD – Python Demo

Input File Name: `mbox-short.txt`



The screenshot shows a text editor window titled 'mbox-short'. The window contains the following text:

```
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008
Return-Path: <postmaster@collab.sakaiproject.org>
Received: from murder (mail.umich.edu [141.211.14.90])
    by frankenstein.mail.umich.edu (Cyrus v2.3.8) with LMTPA;
    Sat, 05 Jan 2008 09:14:16 -0500
X-Sieve: CMU Sieve 2.3
Received: from murder ([unix socket])
    by mail.umich.edu (Cyrus v2.2.12) with LMTPA;
    Sat, 05 Jan 2008 09:14:16 -0500
Received: from holes.mr.itd.umich.edu (holes.mr.itd.umich.edu [141.211.14.79])
    by flawless.mail.umich.edu () with ESMTP id m05EEFR1013674;
    Sat, 5 Jan 2008 09:14:15 -0500
Received: FROM paploo.uhi.ac.uk (app1.prod.collab.uhi.ac.uk [194.35.219.184])
    BY holes.mr.itd.umich.edu ID 477F90B0.2DB2F.12494 ;
    5 Jan 2008 09:14:10 -0500
Received: from paploo.uhi.ac.uk (localhost [127.0.0.1])
    by paploo.uhi.ac.uk (Postfix) with ESMTP id 5F919BC2F2;
    Sat, 5 Jan 2008 14:10:05 +0000 (GMT)
Message-ID: <200801051412.m05ECIaH010327@nakamura.uits.iupui.edu>
Mime-Version: 1.0
Content-Transfer-Encoding: 7bit
```

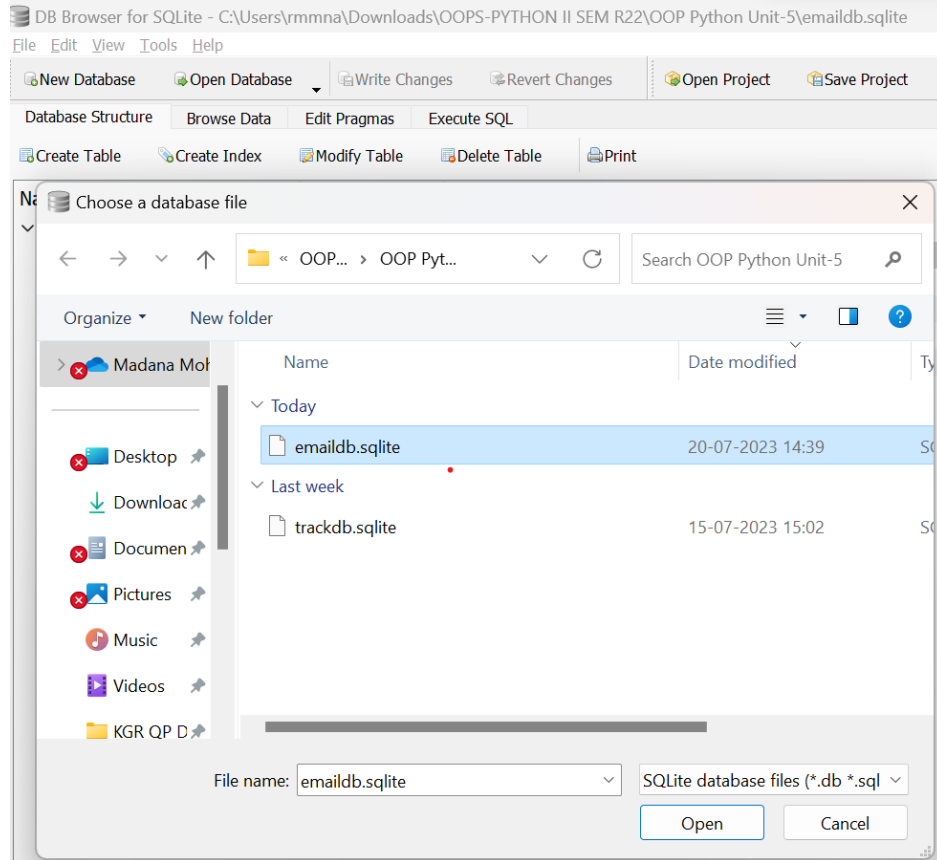
Single Table CRUD – Python Demo

Output:

```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.19
29 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informati
on.
>>>
= RESTART: C:\Users\rmmna\Downloads\OOPS-PYTHON II SEM R22\OOP Python
Unit-5\emaildb.py
Enter file name: mbox-short.txt
cwen@iupui.edu 5
zqian@umich.edu 4
david.horwitz@uct.ac.za 4
louis@media.berkeley.edu 3
gsilver@umich.edu 3
stephen.marquard@uct.ac.za 2
rjlowe@iupui.edu 2
wagnermr@iupui.edu 1
antranig@caret.cam.ac.uk 1
gopal.ramasammycook@gmail.com 1
>>>
```

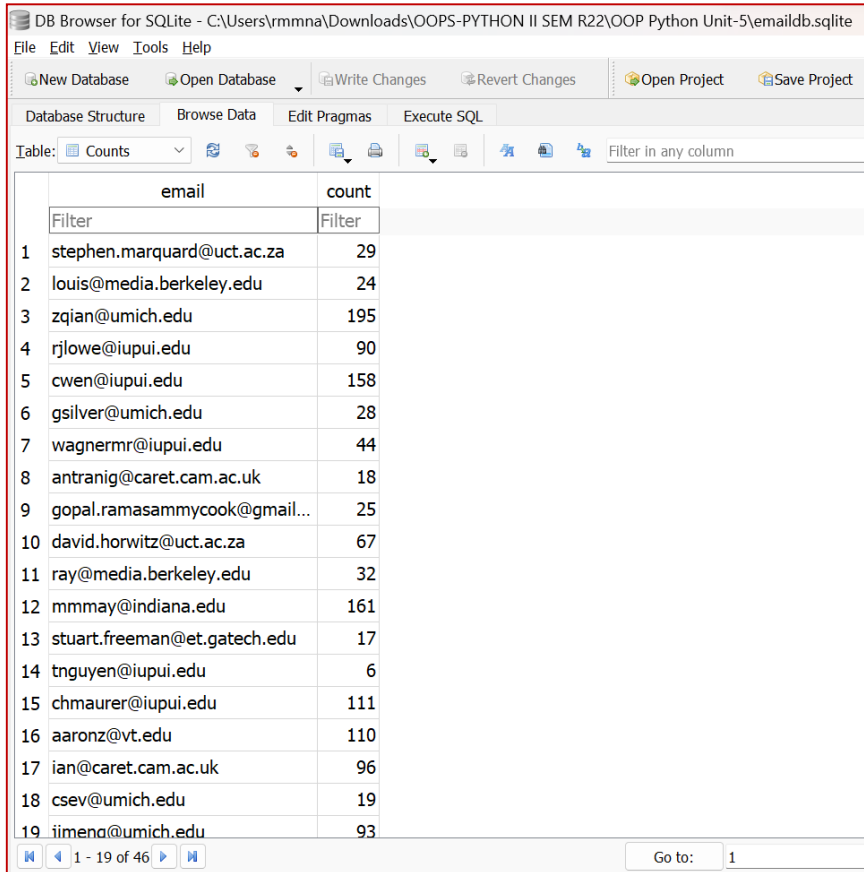
Single Table CRUD – Python Demo

Check the created database `emaildb.sqlite` in DB Browser for SQLite:



Single Table CRUD – Python Demo

Check the created database `emaildb.sqlite` in DB Browser for SQLite:

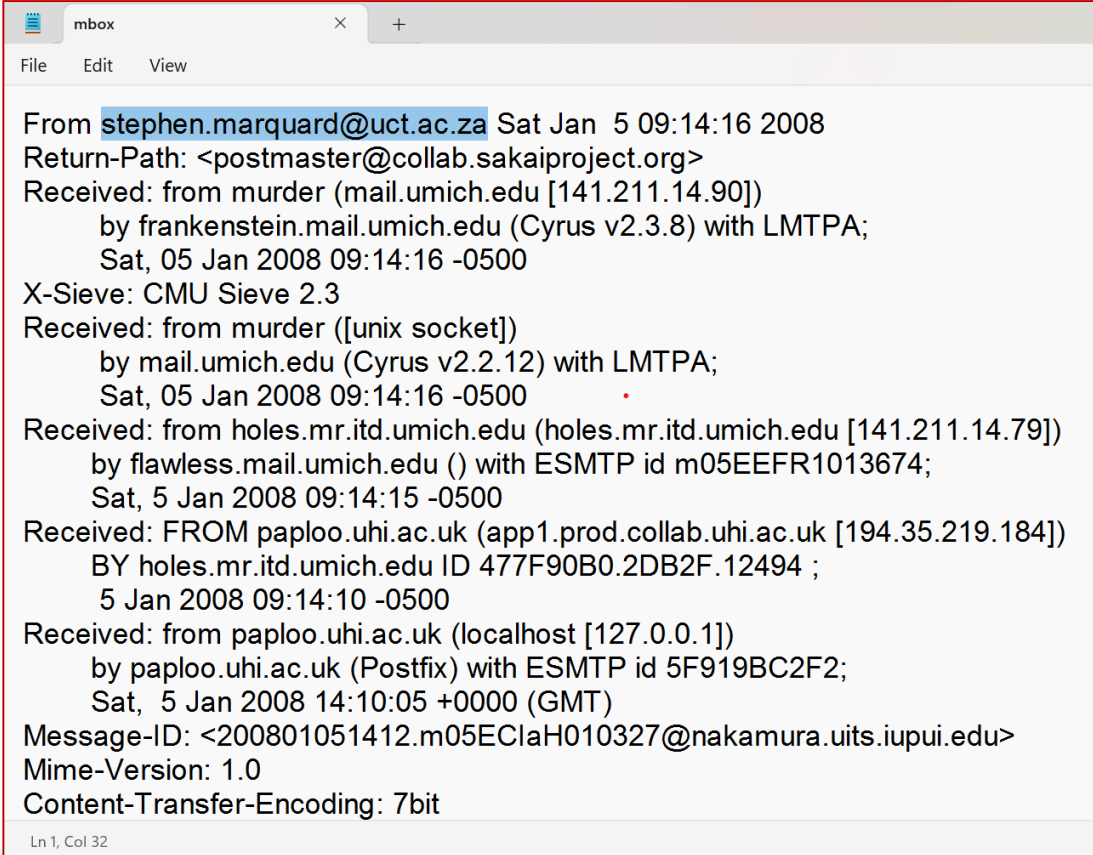


The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database path: `C:\Users\rmmna\Downloads\OOPS-PYTHON II SEM R22\OOP Python Unit-5\emaildb.sqlite`. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains buttons for New Database, Open Database, Write Changes, Revert Changes, Open Project, and Save Project. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The current view is 'Browse Data' for the table 'Counts'. The table has two columns: 'email' and 'count'. The data is displayed in a grid with 19 rows. The first row is a header with 'Filter' in the 'email' column and 'Filter' in the 'count' column. The subsequent rows contain email addresses and their corresponding counts.

	email	count
	Filter	Filter
1	stephen.marquard@uct.ac.za	29
2	louis@media.berkeley.edu	24
3	zqian@umich.edu	195
4	rjlowe@iupui.edu	90
5	cwen@iupui.edu	158
6	gsilver@umich.edu	28
7	wagnermr@iupui.edu	44
8	antranig@caret.cam.ac.uk	18
9	gopal.ramasammycook@gmail...	25
10	david.horwitz@uct.ac.za	67
11	ray@media.berkeley.edu	32
12	mmmay@indiana.edu	161
13	stuart.freeman@et.gatech.edu	17
14	tnguyen@iupui.edu	6
15	chmaurer@iupui.edu	111
16	aaronz@vt.edu	110
17	ian@caret.cam.ac.uk	96
18	csev@umich.edu	19
19	imena@umich.edu	93

Single Table CRUD – Python Demo

Input File Name: mbox.txt



The screenshot shows a text editor window titled 'mbox' with a menu bar containing 'File', 'Edit', and 'View'. The main content area displays an email header with the following text:

```
From: stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008
Return-Path: <postmaster@collab.sakaiproject.org>
Received: from murder (mail.umich.edu [141.211.14.90])
        by frankenstein.mail.umich.edu (Cyrus v2.3.8) with LMTPA;
        Sat, 05 Jan 2008 09:14:16 -0500
X-Sieve: CMU Sieve 2.3
Received: from murder ([unix socket])
        by mail.umich.edu (Cyrus v2.2.12) with LMTPA;
        Sat, 05 Jan 2008 09:14:16 -0500
Received: from holes.mr.itd.umich.edu (holes.mr.itd.umich.edu [141.211.14.79])
        by flawless.mail.umich.edu () with ESMTP id m05EEFR1013674;
        Sat, 5 Jan 2008 09:14:15 -0500
Received: FROM paploo.uhi.ac.uk (app1.prod.collab.uhi.ac.uk [194.35.219.184])
        BY holes.mr.itd.umich.edu ID 477F90B0.2DB2F.12494 ;
        5 Jan 2008 09:14:10 -0500
Received: from paploo.uhi.ac.uk (localhost [127.0.0.1])
        by paploo.uhi.ac.uk (Postfix) with ESMTP id 5F919BC2F2;
        Sat, 5 Jan 2008 14:10:05 +0000 (GMT)
Message-ID: <200801051412.m05EClah010327@nakamura.uits.iupui.edu>
Mime-Version: 1.0
Content-Transfer-Encoding: 7bit
```

At the bottom of the window, the status bar indicates 'Ln 1, Col 32'.

Single Table CRUD – Python Demo

Output:

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\rmmna\Downloads\OOPS-PYTHON II SEM R22\OOP Python Unit-5\emaildb.py
Enter file name: mbox.txt
zqian@umich.edu 195
mmmay@indiana.edu 161
cwen@iupui.edu 158
chmaurer@iupui.edu 111
aaronz@vt.edu 110
ian@caret.cam.ac.uk 96
jimeng@umich.edu 93
rjlowe@iupui.edu 90
dlhaines@umich.edu 84
david.horwitz@uct.ac.za 67
>>>
```