TUS

Topic 3
Build an SQLite Database

Dr Diarmuid Ó Briain

Licence

Module Objectives

At the end of this topic the learner will:

- Build an SQLite database
- Perform SQL queries on the database
- Review the data within the database
- Build a database using the Python sqlite3 module

What is SQLite

- SQLite library
  - A RDBMS contained in a C library
  - SQLite is not a client–server database engine, it is embedded into the end program
  - A popular choice as embedded database software for local/client storage
  - ACID compliant and implements most of the SQL standard
- Sqlite3
  - A terminal-based front-end to the SQLite library that can evaluate queries interactively and display the results in multiple formats
  - sqlite3 can also be used within shell scripts and other applications to provide batch processing features
- DB Browser for SQLite
  - GUI editor for SQLite databases

Install SQLite

- Install the database, the SQLite3 python module and the SQLite Browser

```
~$ sudo apt install -y sqlite3 sqlitebrowser
```


Structured Query Language

- SQL is a specific language used in programming and designed for managing data held in a RDBMS, or for stream processing in a relational data stream management system
- In this case, the RDBMS is SQLite3

Structured Query Language

- Using the **sqlite3** terminal client open a new database

```
~$ sqlite3 db_1.sqlite
```

- Check if there are any current tables in the database

```
sqlite> .tables
```

Create a table

- Create a **class_list** table

```
sqlite> CREATE TABLE IF NOT EXISTS class_list (ref_no INTEGER
PRIMARY KEY, fname TEXT, sname TEXT, number INTEGER);
```

- Confirm the table has been created

```
sqlite> .tables
class_list
```


Insert data into the table

- Add data to the **class_list** table

```
sqlite> INSERT INTO class_list (ref_no, fname, sname, number)
VALUES (0, 'Tom', 'Ryan', 111111);

sqlite> INSERT INTO class_list (ref_no, fname, sname, number)
VALUES (1, 'Mary', 'Murphy', 222222);

sqlite> INSERT INTO class_list (ref_no, fname, sname, number)
VALUES (2, 'Ada', 'Lovelace', 333333);

sqlite> INSERT INTO class_list (ref_no, fname, sname, number)
VALUES (3, 'Charles', 'Babbage', 444444);
```


Insert data into the table

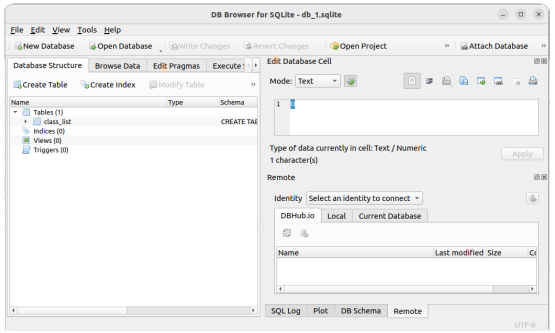
- Confirm the data is in the table via the terminal program

```
sqlite> SELECT * FROM class_list;
0|Tom|Ryan|111111
1|Mary|Murphy|222222
2|Ada|Lovelace|333333
3|Charles|Babbage|444444
```


SQLite Browser

- Run the **sqlitebrowser** to confirm

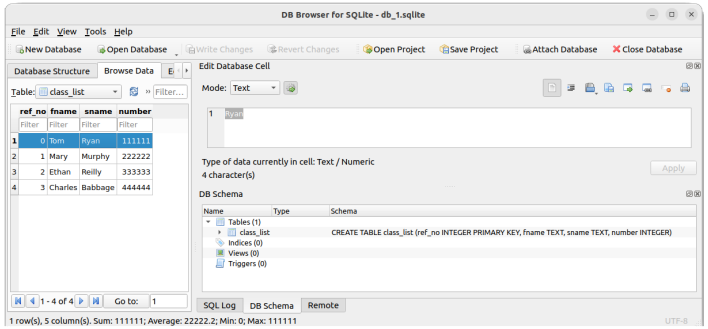
```
ada:~$ sqlitebrowser db_1.sqlite
```



SQLite Browser

- Run the **sqlitebrowser** to confirm

```
ada:~$ sqlitebrowser db_1.sqlite
```



Update data in the table

- Update data in the **class_list** table

```
sqlite> UPDATE class_list SET fname = 'Ethan' WHERE number
LIKE 333333;

sqlite> UPDATE class_list SET sname = 'Reilly' WHERE ref_no
LIKE 2;
```

- Confirm the change

```
sqlite> SELECT * FROM class_list;
0|Tom|Ryan|111111
1|Mary|Murphy|222222
2|Ethan|Reilly|333333
3|Charles|Babbage|444444
```


Select data from the table using Wildcard

- Select data in the **sname** table where name includes "**y**"
  - Note: "%" is the wildcard

```
sqlite> SELECT * FROM class_list WHERE sname LIKE "y";


sqlite> SELECT * FROM class_list WHERE sname LIKE "%y";
1|Mary|Murphy|222222


sqlite> SELECT * FROM class_list WHERE sname LIKE "%y%";
0|Tom|Ryan|111111
1|Mary|Murphy|222222
```


Delete data from the table

- Delete some data from the table. The SQL query deletes all rows from the **class_list** table whose reference number starts with the digit **3**

```
sqlite> DELETE FROM class_list WHERE ref_no LIKE 3;
```

- Confirm the change

```
sqlite> SELECT * FROM class_list;
0|Tom|Ryan|111111
1|Mary|Murphy|222222
2|Ethan|Reilly|333333
```

Delete all data from the table

- Delete all data from the table

```
sqlite> DELETE FROM class_list;
```

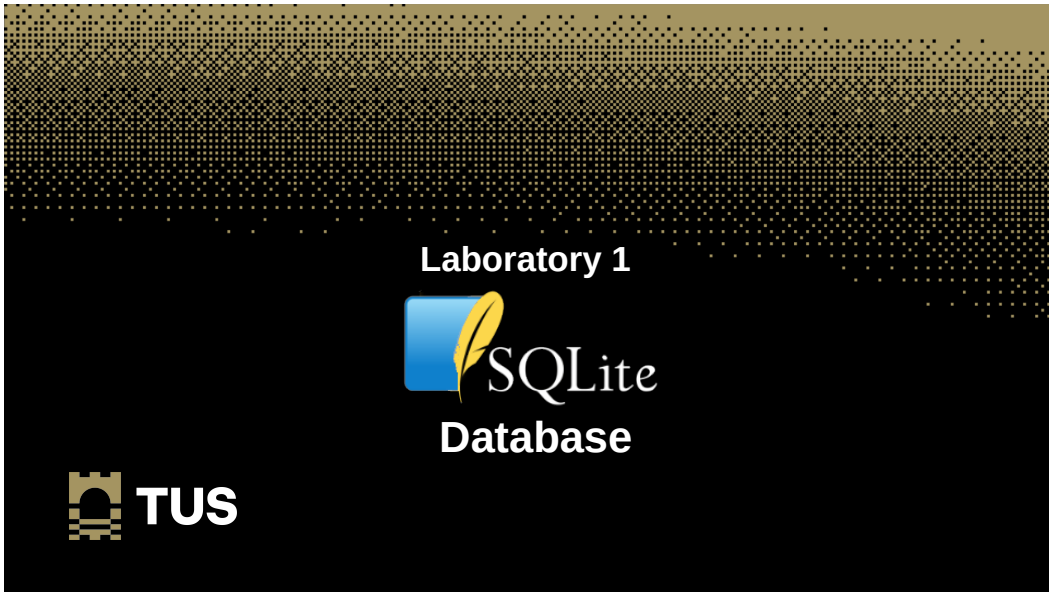
- Confirm the change

```
sqlite> SELECT * FROM class_list;
```


Exit from the database

- Exit from the database

```
sqlite> .quit
```

Laboratory #1

- Write an SQL Database that includes information on some cars outside the window. If you cannot see cars make them up.
- Include:
  - Car manufacturers
  - Car model
  - Car colour
  - Car registration
  - Wheel type, alloy, etc...

Python sqlite3 module

TUS

Interfacing with databases via Python

```
~$ cat working_with_db.py
#! /usr/bin/env python3

import pprint
import sqlite3
import sys

# Defined variables
database = "db_2.sqlite"
table = "class_list"
columns = {
    "ref_no": "INTEGER PRIMARY KEY",
    "fname": "TEXT",
    "sname": "TEXT",
    "number": "INTEGER",
}
data = (
    (0, "Tom", "Ryan", 111111),
    (1, "Mary", "Murphy", 222222),
    (2, "Ada", "Lovelace", 333333),
    (3, "Charles", "Babbage", 444444),
)
list_ = list()
str_ = str()
tuple_ = tuple()

```


Interfacing with databases via Python

```
# // Query Function //
def query_(query):
    """Query function for the Database"""

    list_ = list()
    with sqlite3.connect(database) as con:
        cur = con.cursor()
        cur.execute(query)
        list_ = cur.fetchall()  # Empty except for SELECT
        cur.close()  # Close database cursor
        con.commit()  # Commit labels to the database
    return (0, list_)


# // Use SQL to drop a 'class_list' table if it currently exists //
print(f"Dropping '{table}' from the db '{database}' if it exists")
query_(f"DROP TABLE IF EXISTS {table}")

# // Use SQL to create new 'class_list' table //
print(f"Creating '{table}' in the '{database}' db")
for (key, value) in columns.items():
    list_.append(f"{key} {value}")
str_ = ", ".join(list_)
query_(f"CREATE TABLE IF NOT EXISTS {table} ({str_})")

```


Interfacing with databases via Python

```
# // Get input and put in the database table 'class_list' //
str_ = ", ".join(columns.keys())
for d in data:
    print(f"Inserting {d} into the '{table}' table")
    query_(f"INSERT INTO {table} ({str_}) VALUES {d}")

# // Getting data from database table 'class_list' //
print(f"Retrieving {d} from the '{table}' table")
(_, list_) = query_(f"SELECT * FROM {table}")

# // Printing table 'class_list' //
for t in list_:
    print("    ", ", ".join([str(e) for e in t]))

```

Interfacing with databases via Python

- Running the program

```
~$ ./python_sqlite.py
Dropping 'class_list' from the db 'db_2.sqlite' if it exists
Creating 'class_list' in the 'db_2.sqlite' db
Inserting (0, 'Tom', 'Ryan', 111111) into the 'class_list' table
Inserting (1, 'Mary', 'Murphy', 222222) into the 'class_list' table
Inserting (2, 'Ada', 'Lovelace', 333333) into the 'class_list' table
Inserting (3, 'Charles', 'Babbage', 444444) into the 'class_list' table
Retrieving (3, 'Charles', 'Babbage', 444444) from the 'class_list' table
    0, Tom, Ryan, 111111
    1, Mary, Murphy, 222222
    2, Ada, Lovelace, 333333
    3, Charles, Babbage, 444444
```

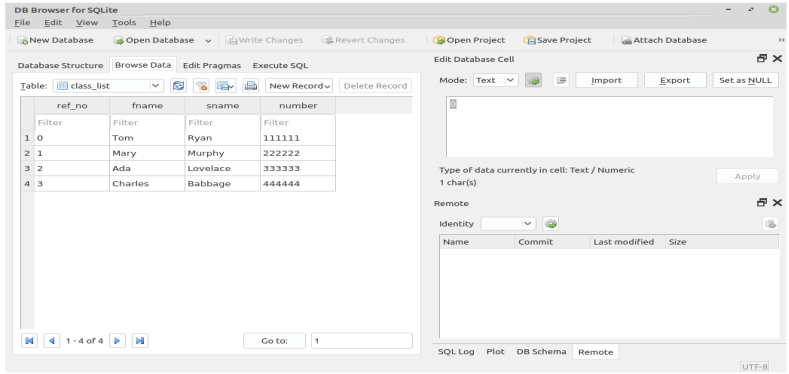

Interfacing with databases via Python

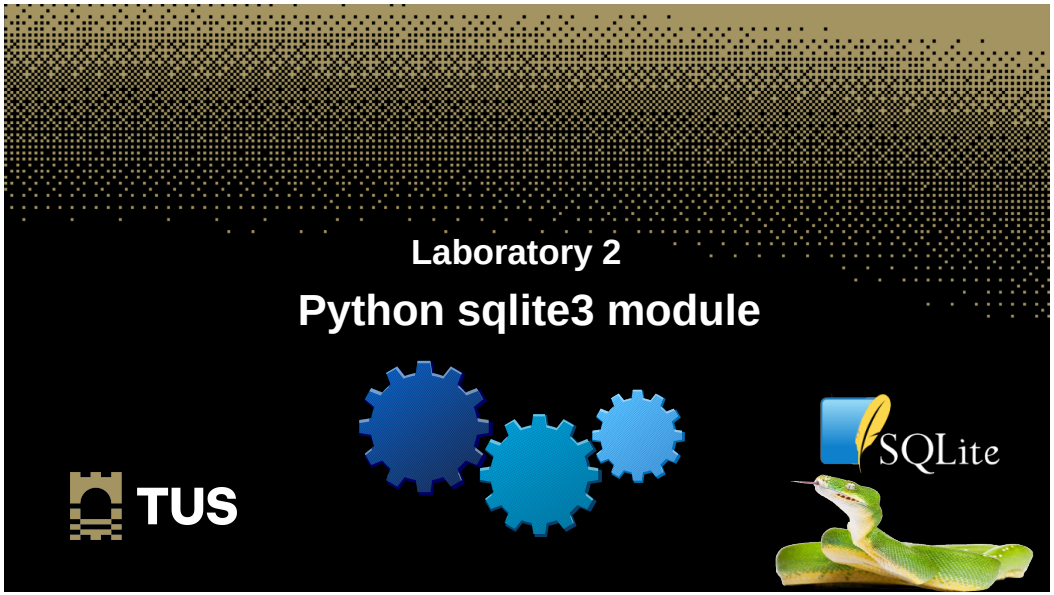
- Checking the table

```
~$ sqlite3 db_2.sqlite
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.

sqlite> .tables
class_list

sqlite> SELECT * FROM class_list;
0|Tom|Ryan|111111
1|Mary|Murphy|222222
2|Ada|Lovelace|333333
3|Charles|Babbage|444444

sqlite> .quit
```


Interfacing with databases

- SQLite Database Browser





Laboratory #2

- Add a shebang line and a document string "**`Exercise #3.2 in Python3`**".
- Import the **`sqlite3`** module and declare global variables.
- Copy the **`query()`** function from the "**`python_sqlite.py`**" program.
- Open the database and retrieve the data from the "**`class_list`**" table.
- Remove "**`Mary Murphy`**" from the course.
- Replace "**`Mary Murphy`**" with her sister "**`Nora`**".
- Add "**`Leo Ashe`**" to the course with a student number "**`555555`**".
- Output the current state of the table.

Laboratory #2

- Output should be somewhat like this:

```
~$ exercise3.2.py
Retrieving data from the 'class_list' table
    0, Tom, Ryan, 111111
    1, Mary, Murphy, 222222
    2, Ada, Lovelace, 333333
    3, Charles, Babbage, 444444
Deleting 'Mary Murphy from the 'class_list'
Inserting 1, Nora, Murphy, 222222 into the 'class_list' table
Inserting 4, Leo, Ashe, 555555 into the 'class_list' table
Retrieving data from the 'class_list' table
    0, Tom, Ryan, 111111
    1, Nora, Murphy, 222222
    2, Ada, Lovelace, 333333
    3, Charles, Babbage, 444444
    4, Leo, Ashe, 555555
```


Learning Objectives

- Build an SQLite database ✓
- Perform SQL queries on the database ✓
- Review the data within the database ✓
- Build a database using the Python sqlite3 module ✓

TUS
Ollscoil Teicneolaíochta na Sionainne
Lár Tíre, An tIarthar Láir
Technological University of the Shannon:
Midlands Midwest

EUR ING Dr Diarmuid Ó Briain
Innealtóir Cairte agus
Léachtóir Sinsearach

E diarmuid.obriain@tus.ie | W tus.ie
Campas Maoilis, Páirc Maoilis,
Luimneach, V94 EC5T, Éire

CEng, FIEI
CISSP®

Thank you

TUS