SQLite 1 Database

CS311: Computer Programming II

ผู้ช่วยศาสตราจารย์สิรินธร จียาศักดิ์

AGENDA

1. What is a Database?
2. What is a Relational DB?
3. SQLite Database Basic

1 What is a Database

PROPOSE OF DATABASE

- A DATABASE SHOULD BE A REPOSITORY OF DATA NEEDED FOR AN ORGANIZATION'S DATA PROCESSING. THAT DATA SHOULD BE ACCURATE, PRIVATE, AND PROTECTED FROM DAMAGE. IT SHOULD BE ACCURATE SO THAT DIVERSE APPLICATIONS WITH DIFFERENT DATA REQUIREMENTS CAN EMPLOY THE DATA. DIFFERENT APPLICATION PROGRAMMERS AND VARIOUS END-USERS HAVE DIFFERENT VIEWS UPON DATA, WHICH MUST BE DERIVED FROM A COMMON OVERALL DATA STRUCTURE. THEIR METHODS OF SEARCHING AND ACCESSING OF DATA WILL BE DIFFERENT.

ADVANTAGES OF USING DATABASE

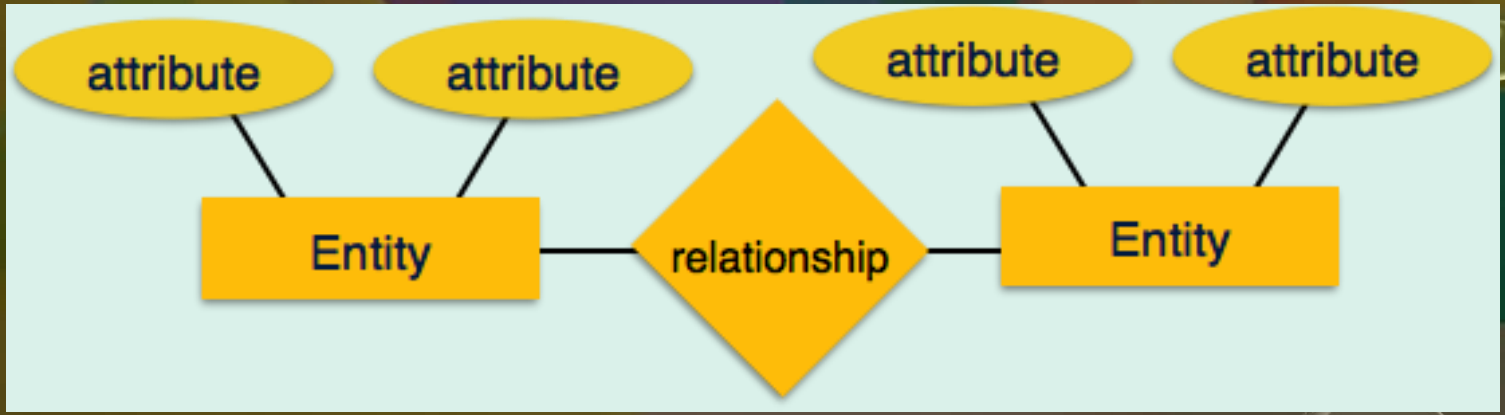
- DATABASE MINIMIZES DATA REDUNDANCY TO A GREAT EXTENT.
- THE DATABASE CAN CONTROL THE INCONSISTENCY OF DATA TO A LARGE EXTENT.
- SHARING OF DATA IS ALSO POSSIBLE USING THE DATABASE.
- DATABASE ENFORCE STANDARDS.
- THE USE OF DATABASES CAN ENSURE DATA SECURITY.
- INTEGRITY CAN BE MANAGED USING THE DATABASE.

ADVANTAGES OF USING DATABASE

- DATA MODELS DEFINE HOW THE LOGICAL STRUCTURE OF A DATABASE IS MODELED. DATA MODELS ARE FUNDAMENTAL ENTITIES TO INTRODUCE ABSTRACTION IN A DBMS. DATA MODELS DEFINE HOW DATA IS CONNECTED TO EACH OTHER AND HOW THEY ARE PROCESSED AND STORED INSIDE THE SYSTEM
  - ENTITY-RELATIONSHIP MODEL (ER MODEL)
  - RELATIONAL MODEL

ENTITY-RELATIONSHIP MODEL

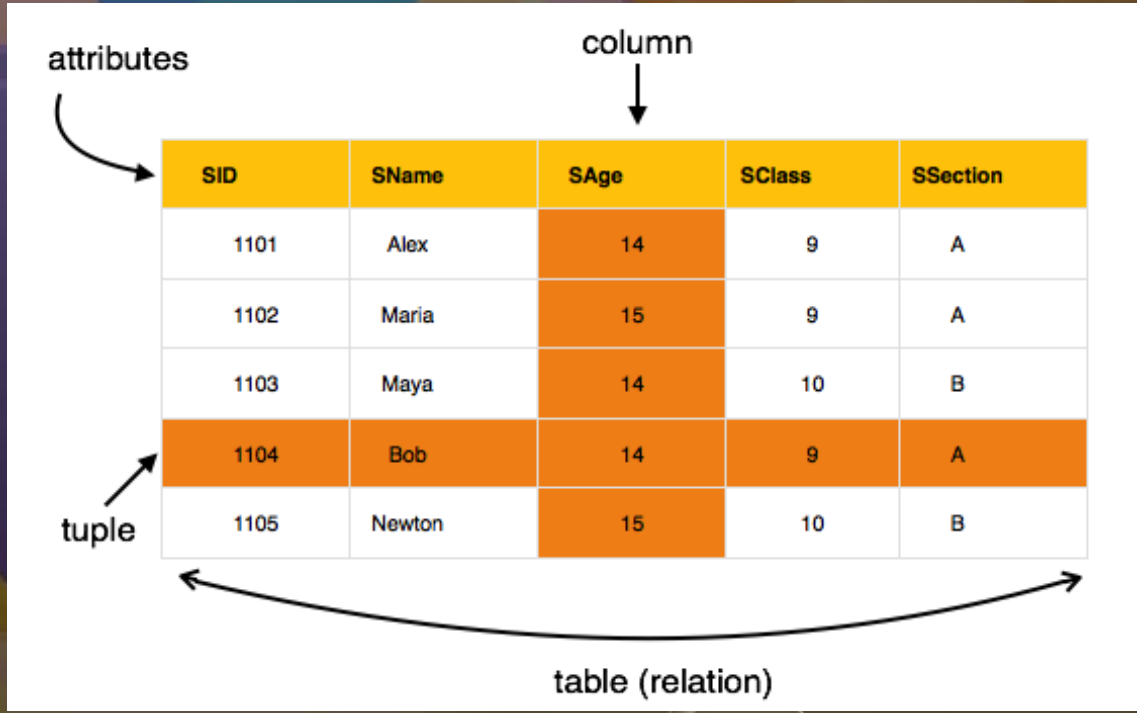
- **ER MODEL** IS BEST USED FOR THE CONCEPTUAL DESIGN OF A DATABASE
- ER MODEL IS BASED ON :
  - **ENTITIES** AND THEIR ***ATTRIBUTES***
  - **RELATIONSHIPS** AMONG ENTITIES



https://www.tutorialspoint.com/dbms/dbms_architecture.htm

RELATIONAL MODEL

- THE **MOST POPULAR DATA MODEL IN DBMS IS THE RELATIONAL MODEL**. IT IS MORE SCIENTIFIC A MODEL THAN OTHERS. THIS MODEL IS BASED ON FIRST-ORDER PREDICATE LOGIC AND DEFINES A TABLE AS AN **N-ARY RELATION.**



SID	SName	SAge	SClass	SSection
1101	Alex	14	9	A
1102	Maria	15	9	A
1103	Maya	14	10	B
1104	Bob	14	9	A
1105	Newton	15	10	B

https://www.tutorialspoint.com/dbms/dbms_architecture.htm

2 What is a RDBMS

WHAT IS RDBMS

- RDBMS STANDS FOR
  RELATIONAL DATABASE MANAGEMENT SYSTEM.
  RDBMS IS THE BASIS FOR SQL, AND FOR ALL MODERN DATABASE SYSTEMS LIKE MS SQL SERVER, IBM DB2, ORACLE, MYSQL, AND MICROSOFT ACCESS.
- A RELATIONAL DATABASE MANAGEMENT SYSTEM **(RDBMS)** IS A DATABASE MANAGEMENT SYSTEM (DBMS) THAT IS BASED ON THE RELATIONAL MODEL AS INTRODUCED BY E. F. CODD.

https://www.tutorialspoint.com/dbms/dbms_architecture.htm

WHAT IS RDBMS

- THE DATA IN AN RDBMS IS STORED IN DATABASE OBJECTS WHICH ARE CALLED AS **TABLES**. THIS TABLE IS BASICALLY A COLLECTION OF RELATED DATA **ENTRIES** AND IT CONSISTS OF NUMEROUS COLUMNS AND ROWS.
- REMEMBER, A TABLE IS THE MOST COMMON AND SIMPLEST FORM OF DATA STORAGE IN A RELATIONAL DATABASE. THE FOLLOWING PROGRAM IS AN EXAMPLE OF A **CUSTOMERS TABLE** –

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm

WHAT IS A FIELD/COLUMN

- **FIELD/COLUMN**
  - A FIELD IS A COLUMN IN A TABLE THAT IS DESIGNED TO MAINTAIN SPECIFIC INFORMATION ABOUT EVERY RECORD IN THE TABLE.
  - A COLUMN IS A **VERTICAL ENTITY** IN A TABLE THAT CONTAINS ALL INFORMATION ASSOCIATED WITH A SPECIFIC FIELD IN A TABLE.
  - FOR EXAMPLE, A COLUMN IN THE CUSTOMERS TABLE IS **ADDRESS**, WHICH REPRESENTS LOCATION DESCRIPTION AND WOULD BE AS SHOWN BELOW

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm

WHAT IS A FIELD/COLUMN

- THE CUSTOMER TABLE CONSIST OF 5 FIELDS/COLUMNS

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm

WHAT IS A RECORD OR A ROW

- A **RECORD** IS ALSO CALLED AS A ROW OF DATA IS EACH INDIVIDUAL ENTRY THAT EXISTS IN A TABLE. FOR EXAMPLE,

  THERE ARE **7 RECORDS** IN THE ABOVE CUSTOMERS TABLE. FOLLOWING IS A SINGLE ROW OF DATA OR RECORD IN

  THE **CUSTOMERS TABLE** –

1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm

THE FORMAL DATABASE TERMS

- THE FOLLOWING TABLE LISTS SOME OF THE FORMAL AND INFORMAL RELATIONAL DATABASE TERMS DESCRIBING TABLES AND THEIR CONTENTS, TOGETHER WITH THEIR EQUIVALENT IN OTHER NONRELATIONAL DATABASES. THIS MANUAL USES THE INFORMAL TERMS

Formal relational term	**Informal** relational term	**Equivalent** nonrelational term
Relation	Table	File
Attribute	Column	Field
Tuple	Row	Record

http://www.upi.pr.it/docs/easfg/easvrfgp7.htm

RELATIONAL DATABASE EXAMPLE

- IN RELATIONAL DATABASE, DATA IS STORED IN TABLES. A TABLE CONSISTS OF ROWS AND COLUMNS

DB_CS311

student		
sid	sname	syear
s001	Student1	1
s002	Student2	2
s003	Student3	1

teacher		
tid	tname	tyear
t001	Teacher1	5
t002	Teaher2	6
t003	Teacher3	5

class		
cid	cname	price
c001	class1	3500
c002	class2	4500
c003	class3	5500

https://www.slideshare.net/1keydata/sql-tutorial-table-management

PRIMARY AND FOREIGN KEYS

- **PRIMARY** AND **FOREIGN KEYS** DEFINE THE RELATIONAL STRUCTURE OF A DATABASE. THESE KEYS ENABLE EACH ROW IN THE DATABASE TABLES TO BE IDENTIFIED AND DEFINE THE RELATIONSHIPS BETWEEN THE TABLES.
- ALL TABLES IN A RELATIONAL DATABASE SHOULD HAVE A **PRIMARY KEY**.
- THE PRIMARY KEY IS A COLUMN, OR SET OF COLUMNS, THAT ALLOWS EACH ROW IN THE TABLE TO BE **UNIQUELY** IDENTIFIED. NO TWO ROWS IN A TABLE WITH A PRIMARY KEY CAN HAVE THE SAME PRIMARY KEY VALUE.

https://www.slideshare.net/1keydata/sql-tutorial-table-management

QUERY LANGUAGE

- **SQL** STANDS FOR **STRUCTURED QUERY LANGUAGE**
- **SQL** IS A DATABASE COMPUTER LANGUAGE DESIGNED FOR THE RETRIEVAL AND MANAGEMENT OF DATA IN A RELATIONAL DATABASE.

Select

Update

Insert

Delete

https://www.tutorialspoint.com/sql/index.htm

3 What is a RDBMS

SQLITE IS A SOFTWARE LIBRARY THAT IMPLEMENTS A SELF-CONTAINED, SERVERLESS, ZERO-CONFIGURATION, TRANSACTIONAL SQL DATABASE ENGINE. SQLITE IS THE MOST **WIDELY DEPLOYED SQL DATABASE ENGINE IN THE WORLD**. THE SOURCE CODE FOR SQLITE IS IN THE PUBLIC DOMAIN. THIS TUTORIAL WILL GIVE YOU A QUICK START WITH SQLITE AND MAKE YOU COMFORTABLE WITH SQLITE PROGRAMMING.

https://www.tutorialspoint.com/sql/index.htm

WHY SQLITE ?

- SQLITE DOES NOT REQUIRE A SEPARATE SERVER PROCESS OR SYSTEM TO OPERATE **(SERVERLESS).**
- SQLITE COMES WITH **ZERO-CONFIGURATION**, WHICH MEANS NO SETUP OR ADMINISTRATION NEEDED.
- A COMPLETE SQLITE DATABASE IS STORED IN A SINGLE **CROSS-PLATFORM** DISK FILE.
- SQLITE IS **VERY SMALL AND LIGHT WEIGHT, LESS THAN 400KIB** FULLY CONFIGURED OR LESS THAN 250KIB WITH OPTIONAL FEATURES OMITTED.
- SQLITE IS SELF-CONTAINED, WHICH MEANS NO EXTERNAL DEPENDENCIES. SQLITE TRANSACTIONS ARE **FULLY ACID-COMPLIANT**, ALLOWING SAFE ACCESS FROM MULTIPLE PROCESSES OR THREADS.
- SQLITE **SUPPORTS** MOST OF THE **QUERY LANGUAGE** FEATURES FOUND IN SQL92 (SQL2) STANDARD.
- SQLITE IS WRITTEN IN **ANSI-C** AND PROVIDES SIMPLE AND EASY-TO-USE API.
- SQLITE IS **AVAILABLE** ON **UNIX** (LINUX, MAC OS-X, ANDROID, IOS) AND **WINDOWS** (WIN32, WINCE, WINRT).

https://www.tutorialspoint.com/sql/index.htm

SQLITE – FOR WINDOW

STEP 1 – GO TO (HTTPS://SQLITE.ORG/DOWNLOAD.HTML), AND DOWNLOAD PRECOMPILED BINARIES FROM WINDOWS SECTION.

STEP 2 – DOWNLOAD SQLITE-SHELL-WIN32-*.ZIP AND SQLITE-DLL-WIN32-*.ZIP ZIPPED FILES.

Precompiled Binaries for Mac OS X (x86)

sqlite-tools-osx-x64-3490100.zip (3.54 MiB)	A bundle of command-line tools for managing SQLite database files, including (1) the command-line shell, (2) sqldiff, (3) sqlite3_analyzer, and (4) sqlite3_rsync. (64-bit) (SHA3-256: a5789ccfbc2bac1c4599704f4be68a92d990ccf8f0ef2fbcc39ac3d93131c0e4)

Precompiled Binaries for Windows

sqlite-dll-win-x86-3490100.zip (1.02 MiB)	32-bit DLL (x86) for SQLite version 3.49.1. (SHA3-256: 1dac243f154828327d6a250b74cf115e4f6a4cbfe83dbc488d45510b59e7a157)
sqlite-dll-win-x64-3490100.zip (1.28 MiB)	64-bit DLL (x64) for SQLite version 3.49.1. (SHA3-256: ec8fb7976d9c4bc495c4a142da05e97c4d6dc6c1205877adcce0bd5b191026e2)
sqlite-tools-win-x64-3490100.zip (6.12 MiB)	A bundle of command-line tools for managing SQLite database files, including (1) the command-line shell, (2) sqldiff.exe, (3) sqlite3_analyzer.exe, and (4) sqlite3_rsync.exe. 64-bit. (SHA3-256: f80860a651026229c8d6ba76cd8c3af74e114cecf71ad9fc5bf938adcd4ca0b9)

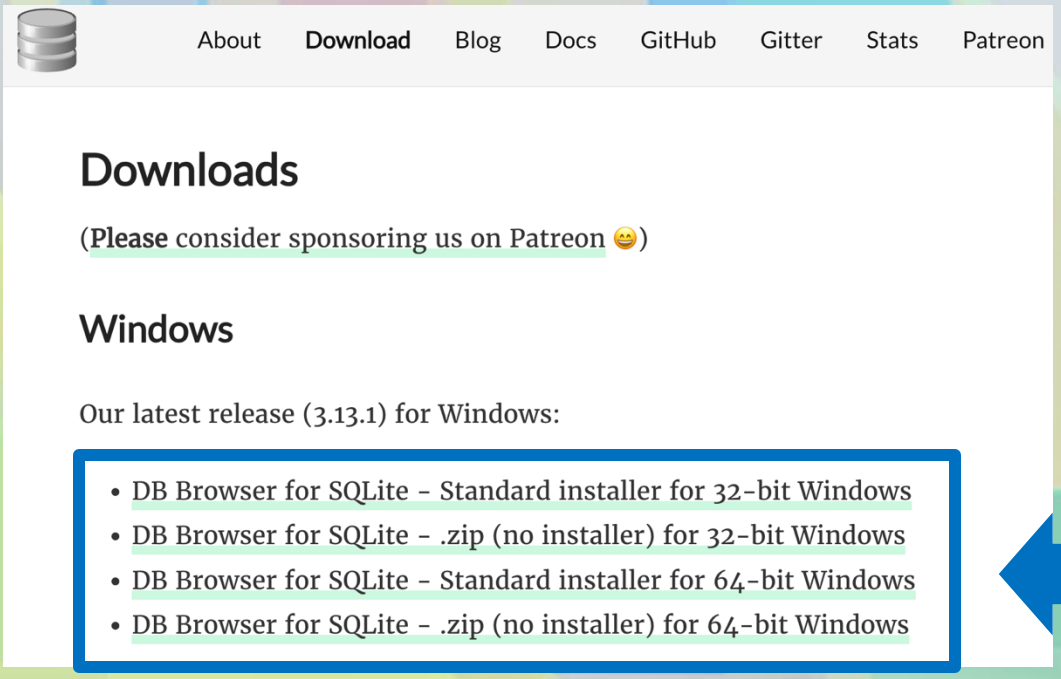
STEP 3 – UNZIP THIS ZIPFILE AND RUN SQLITE3.EXE

Install DB Browser

DB BROWSER – FOR WINDOW

STEP 1 – GO TO (HTTPS://SQLITEBROWSER.ORG), AND DOWNLOAD PRECOMPILED BINARIES FROM WINDOWS SECTION.

STEP 2 – DOWNLOAD DB BROWSER FOR SQLITE

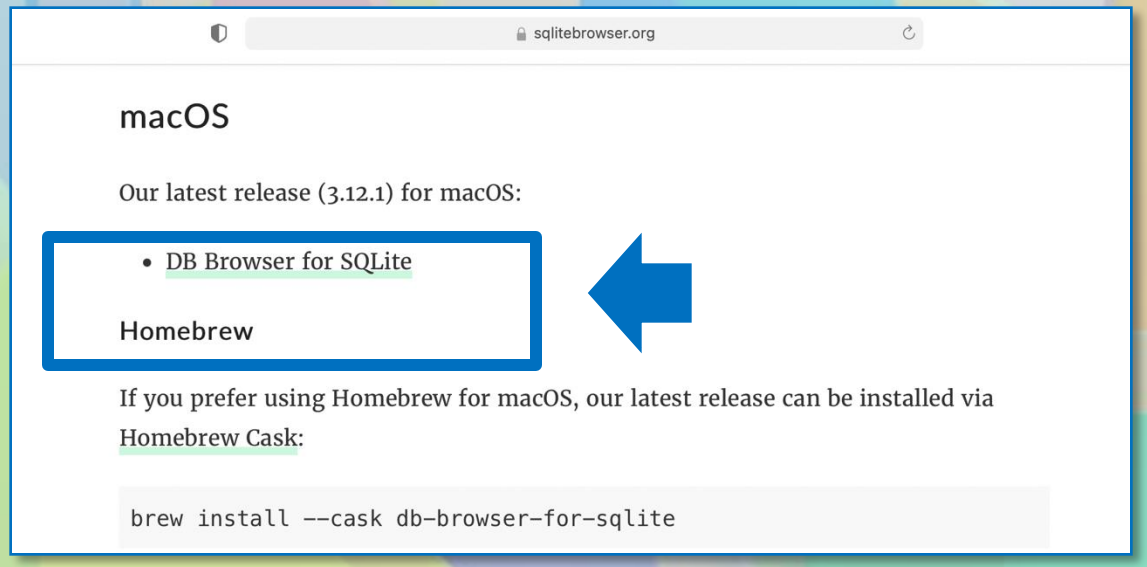


STEP 3 – INSTALL THE FILE NAME DB.BROWSER.FOR.SQLITE-3.12.1-V2.DMG

DB BROWSER – FOR MAC

STEP 1 – GO TO (HTTPS://SQLITEBROWSER.ORG), AND DOWNLOAD PRECOMPILED BINARIES FROM MACOS SECTION.

STEP 2 – DOWNLOAD DB BROWSER FOR SQLITE



STEP 3 – INSTALL THE FILE NAME DB BROWSER FOR SQLITE

STEP 1: INSTALLING DB BROWSER

db browser for sqlite

All | Images | Videos | Shopping | Short videos | Forums | Web | More | Tools

DB Browser for SQLite
https://sqlitebrowser.org

DB Browser for SQLite

DB Browser for SQLite (DB4S) is a high quality, visual, open source tool designed for people who want to create, search, and edit SQLite or SQLCipher database ...

Downloads

Windows · DB Browser for SQLite - Standard installer for 32-bit ...

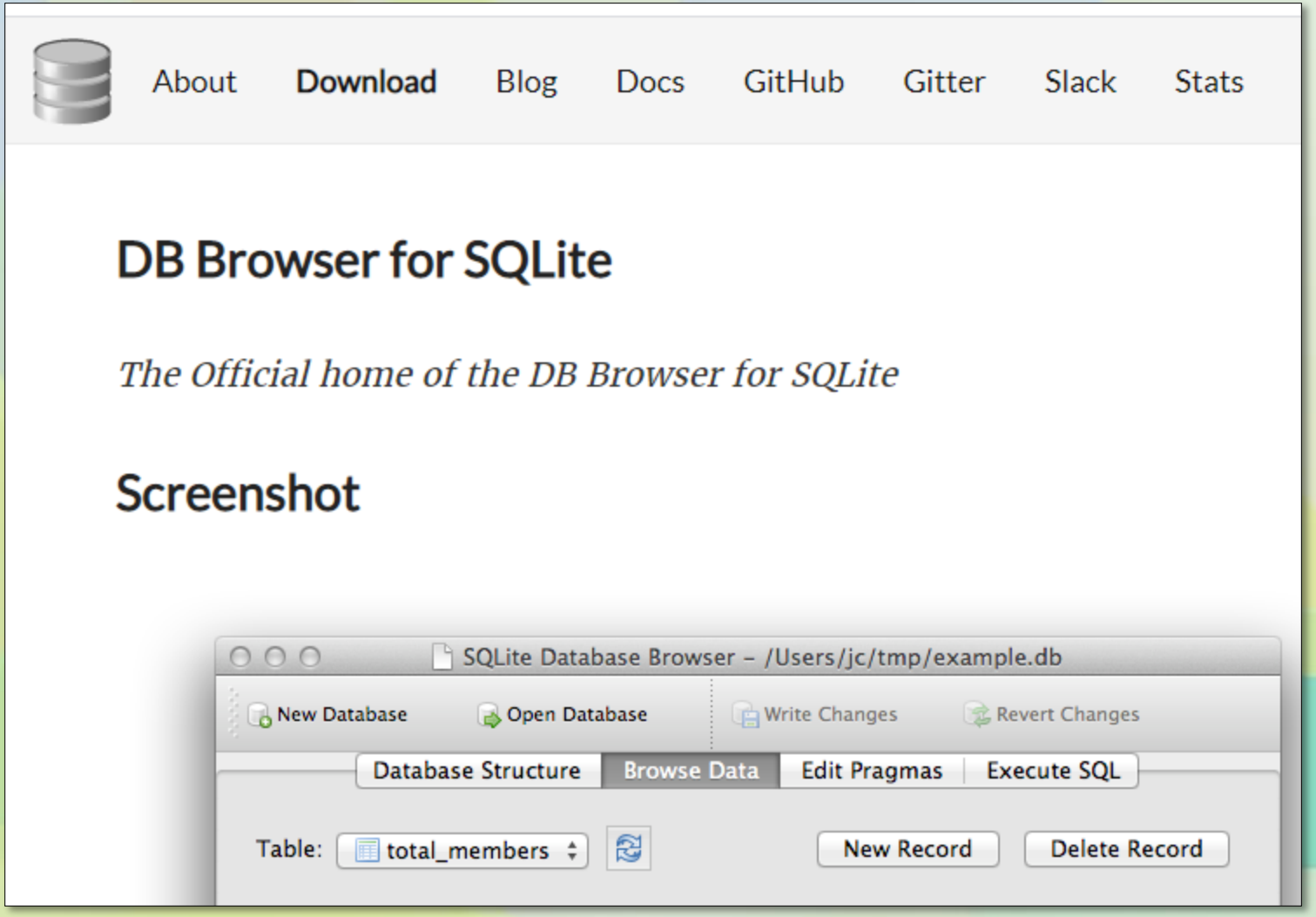
Newdash.io

The next home for your Redash hosting. Please register your ...

Version 3.13.1 released

Downloads · DB.Browser.for.SQLite-v3.13.1-win32.msi ...

STEP2: INSTALLING DB BROWSER



STEP3: INSTALLING DB BROWSER

About | **Download** | Blog | Docs | GitHub | Gitter | Stats | Patreon

Downloads

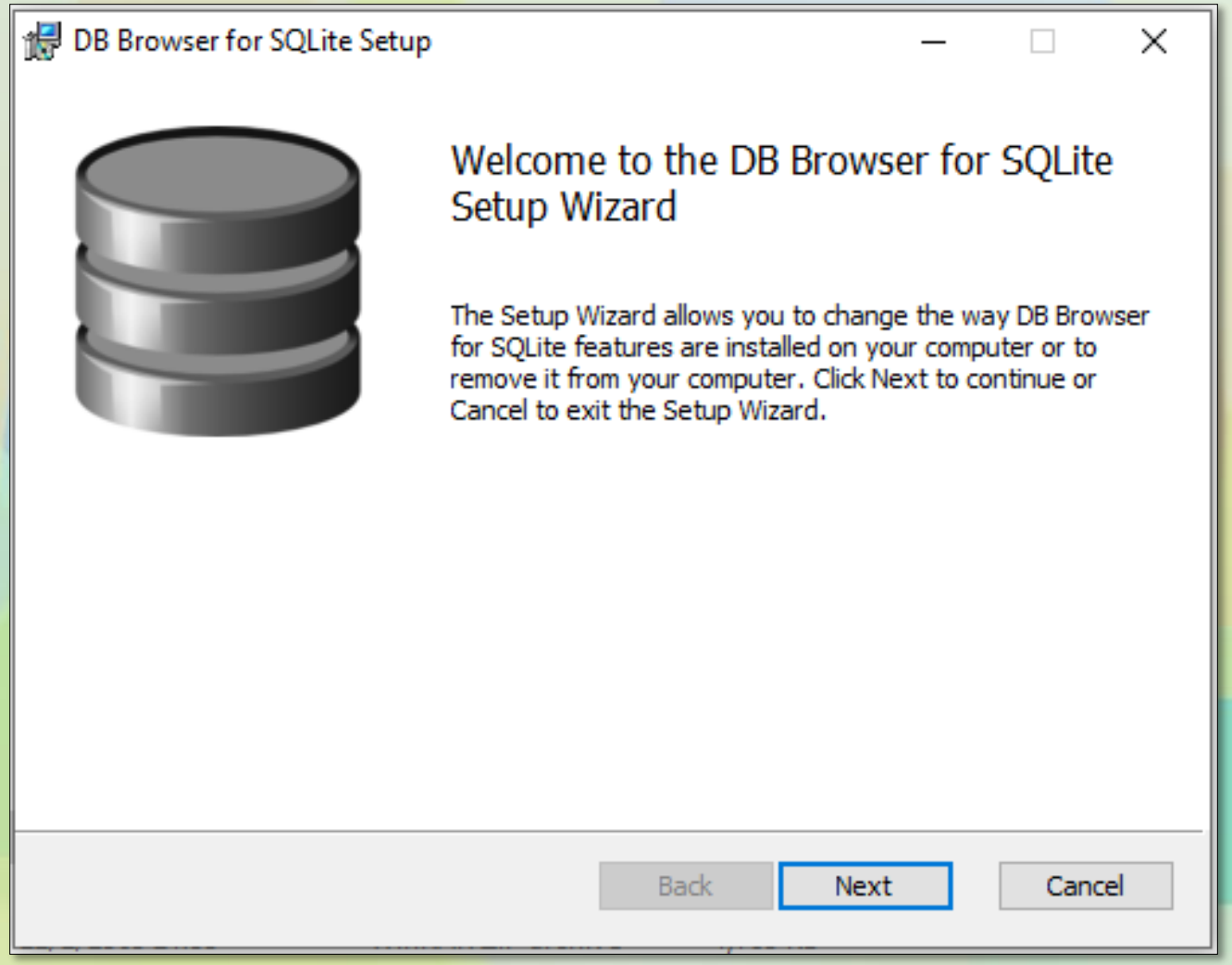
(**Please** consider sponsoring us on Patreon 😄)

Windows

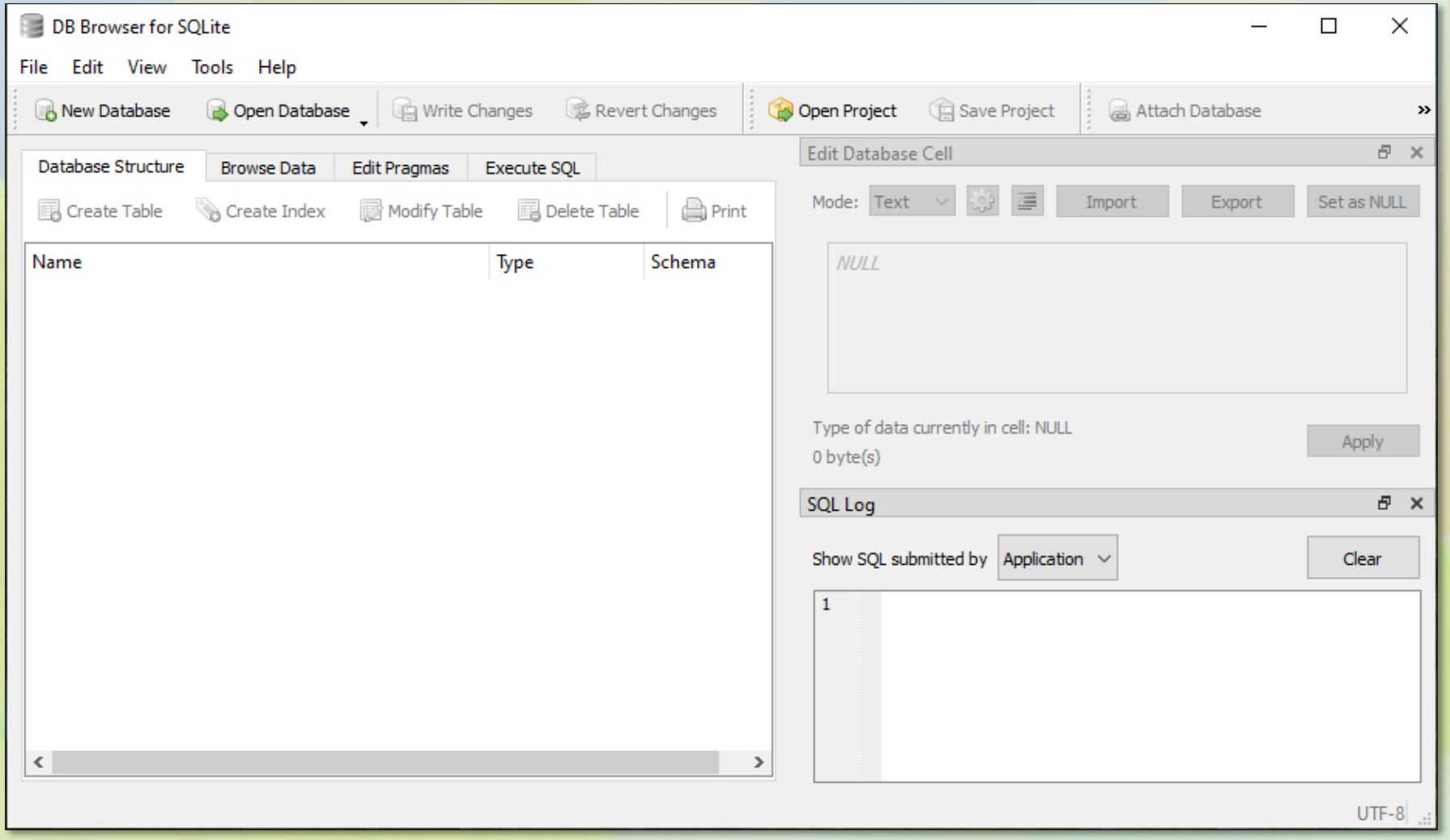
Our latest release (3.13.1) for Windows:

- DB Browser for SQLite - Standard installer for 32-bit Windows
- DB Browser for SQLite - .zip (no installer) for 32-bit Windows
- DB Browser for SQLite - Standard installer for 64-bit Windows
- DB Browser for SQLite - .zip (no installer) for 64-bit Windows

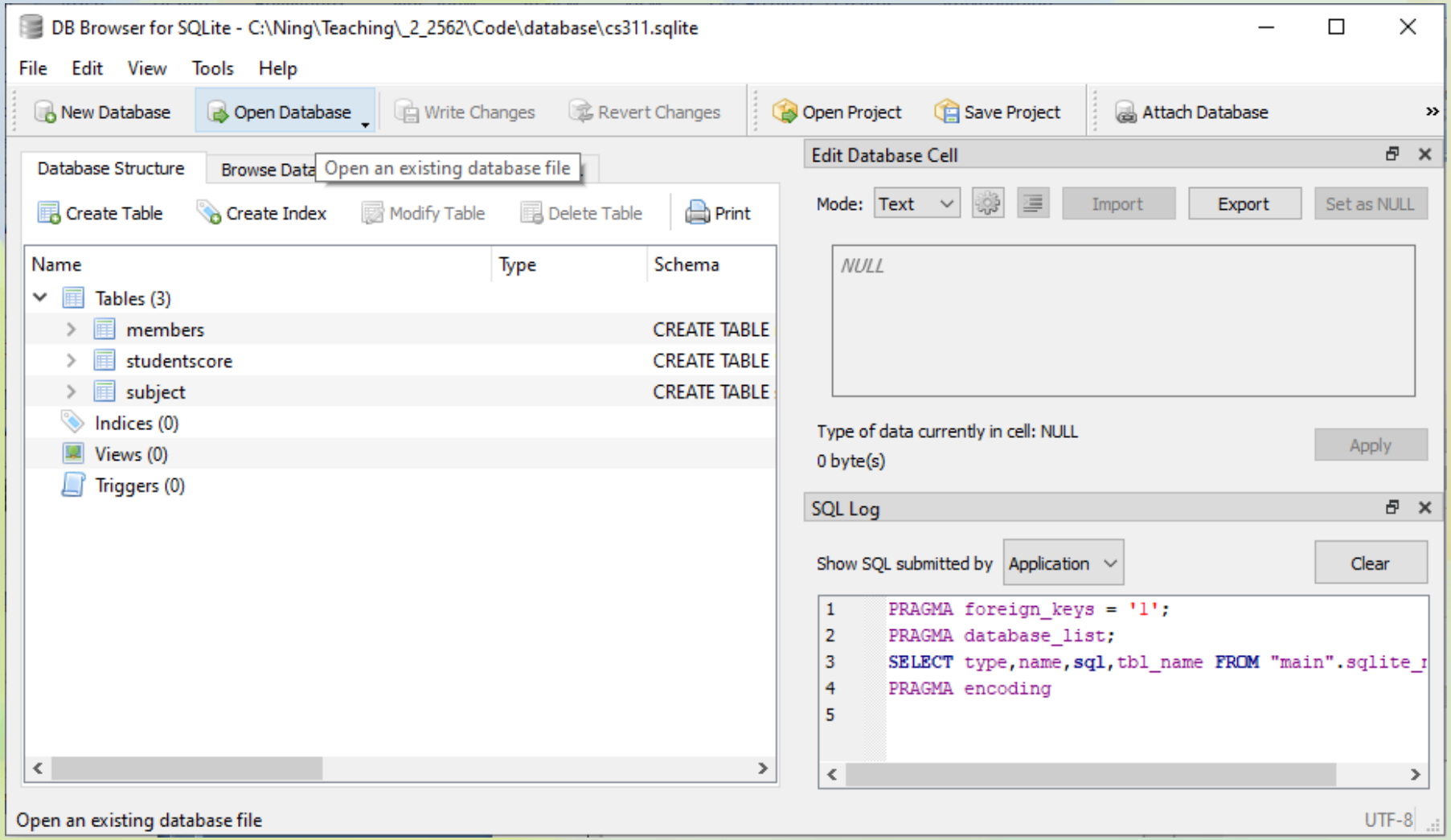
STEP4: INSTALLING DB BROWSER



WELCOME TO DB BROWSER

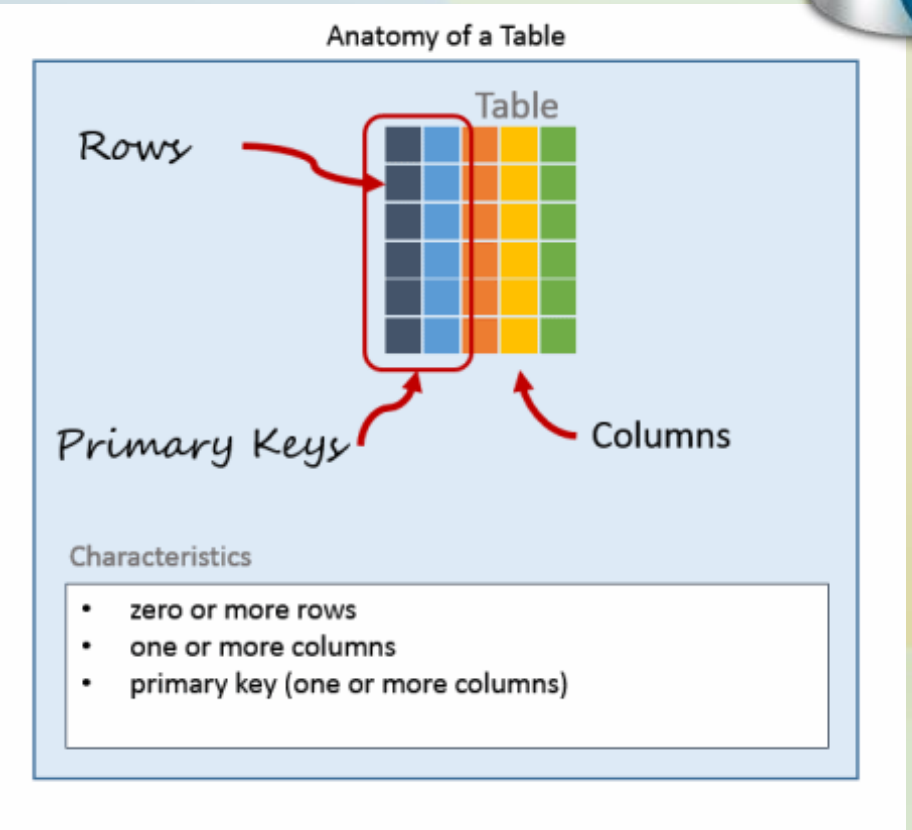


DB BROWSER APPLICATION



CRATING DATABASE STEPS

1. CREATE A **DATABASE**
2. CREATE **TABLES**
   (USING DB BROWSER)
3. **INSERT DATA** TABLE
   (USING DB BROWSER)



Class Activity of Week9

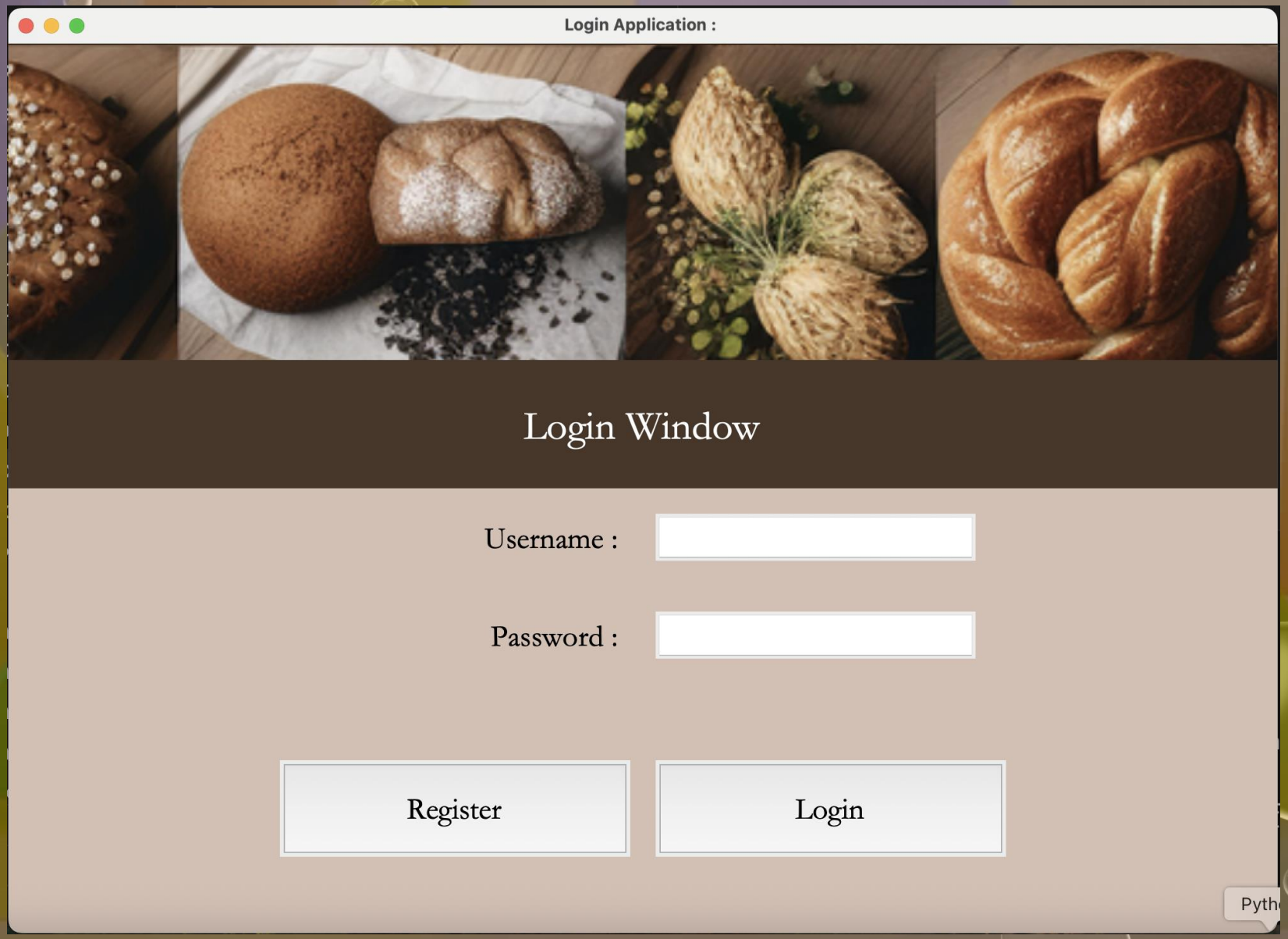
CLASS ACTIVITY : CREATING A DATA TABLES

DB_YOUR STUDENT ID

member		
mid	mname	mage
1001	sirinthorn1	43
1002	sirinthorn2	25
1003	sirinthorn3	27

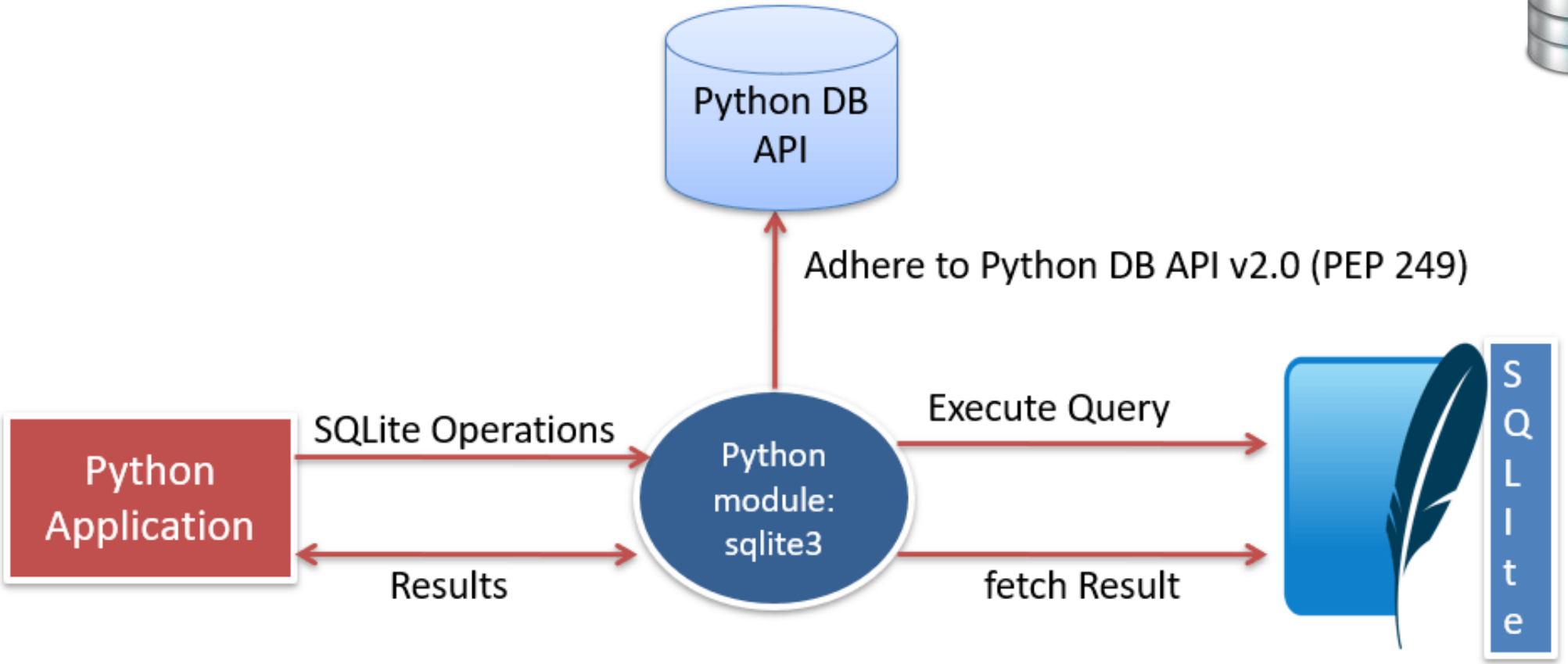
menu		
meid	mename	meprice
1	cake	125
2	drink	85
3	noodle	55

CLASS ACTIVITY : BUILD LOGIN APPLICATION



1 Establishing a Connection

PYTHON SQLITE CONNECTION ARCHITECTURE



https://pynative.com/python-sqlite/

ESTABLISHING CONNECTION USING PYTHON

To establish a connection with SQLite3 database using python :

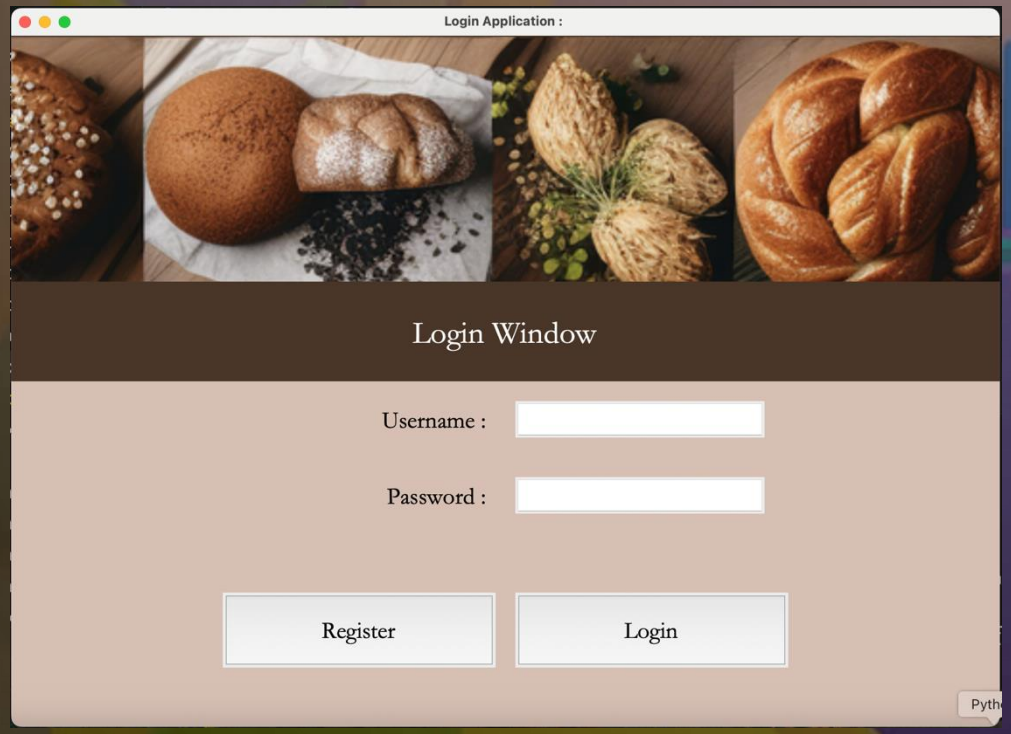
- Import the sqlite3 module using the import statement.
- The connect() method accepts the name of the database you need to connect with as a parameter and, returns a Connection object

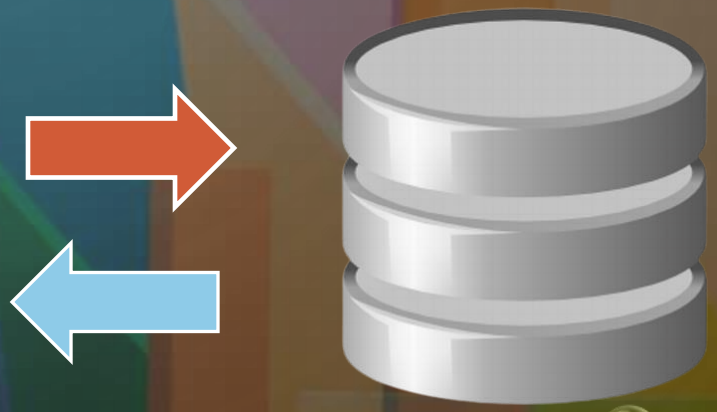
ESTABLISHING CONNECTION EXAMPLE

```
import sqlite3
#Connecting to sqlite
db_file = 'files/testdb1.sqlite'
conn = sqlite3.connect(db_file)

# Commit your changes in the database
conn.commit()

#Closing the connection
conn.close()
```

2 Retrieving data from SQLite

RETRIEVING DATA FROM SQLITE





RETRIEVING DATA FROM SQLITE STEPS USING PYTHON

1. **Import sqlite3 module.** Using the classes and methods defined in the sqlite3 module we can communicate with the SQLite database.
2. **Use the connect() method.** Use the connect() method of the connector class with the database name. To establish a connection to SQLite, you need to pass the database name you want to connect. This method returns the SQLite Connection Object if the connection is successful.
3. **Use the cursor() method.** Use the cursor() method of a connection class to create a cursor object to execute SQLite command/queries from Python.
4. **Use the execute() method.** The execute() methods run the SQL query and return the result.
5. **Extract result using fetchall().** Use cursor.fetchall() or fetchone() or fetchmany() to read query result.
6. **Close cursor and connection objects** use cursor.close() and connection.close() method to close the cursor and SQLite connections after your work completes

https://pynative.com/python-sqlite/

RETRIEVING DATA FROM A DATABASE

- The **SELECT statement** is used to select data from a database. The data returned is stored in a result table, called the result-set

- Syntax :

```
SELECT * FROM table_name;
```

```
SELECT column1, column2, ...
FROM table_name;
```

RETRIEVING DATA FROM A DATABASE

The fetch() method provided by the sqlite python module.

The sqlite3. Cursor class provides three methods namely fetchall(), fetchmany() and, fetchone() where,

- The **fetchall()** method retrieves all the rows in the result set of a query and returns them as list of tuples. (If we execute this after retrieving few rows it returns the remaining ones).
- The **fetchone()** method fetches the next row in the result of a query and returns it as a tuple.
- The **fetchmany()** method is similar to the fetchone() but, it retrieves the next set of rows in the result set of a query, instead of a single row.

RETRIEVING DATA USING FETCHALL()

- FETCHALL() METHOD

```
def select_table(conn):
    cursor = conn.cursor()
    sql = "select * from employee"
    cursor.execute(sql)

    #Fetching all of row that remain from fetch command before
    result = cursor.fetchall()
    amt = len(result)          #return number of row
    print("Total row = ",amt)
    for i,row in enumerate(result) :
        print('Row ',i," ", row)
```

RETRIEVING DATA USING FETCHONE()

- FETCHONE() METHOD

```
def select_table(conn):
    cursor = conn.cursor()
    sql = "select * from employee"
    cursor.execute(sql)

    #Fetching 1st row from the table
    result = cursor.fetchone()
    print(result)
```

RETRIEVING DATA USING FETCHMANY()

- FETCHMANY() METHOD

```
def select_table(conn):
    cursor = conn.cursor()
    sql = "select * from employee"
    cursor.execute(sql)

    #Fetching 1st row from the table
    result = cursor.fetchone()
    print(result)

    result = cursor.fetchmany(2)
    print(result)
```

3 SQL Statement

Select using where clause

SELECT DATA USING WHERE CLAUSE

- The WHERE clause is used to filter records. It is used to extract only those records that fulfill a specified condition
- Syntax :

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

WHERE CLAUSE STATEMENT EXAMPLE

```
SELECT * FROM Customers
WHERE Country='Mexico';
```

```
SELECT * FROM Customers
WHERE CustomerID=1;
```

THE SQL AND, OR AND NOT OPERATORS

The WHERE clause can be combined with AND, OR, and NOT operators.

- The AND and OR operators are used to filter records based on more than one condition:
- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.

SYNTAX FOR USING AND/OR OPERATOR

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

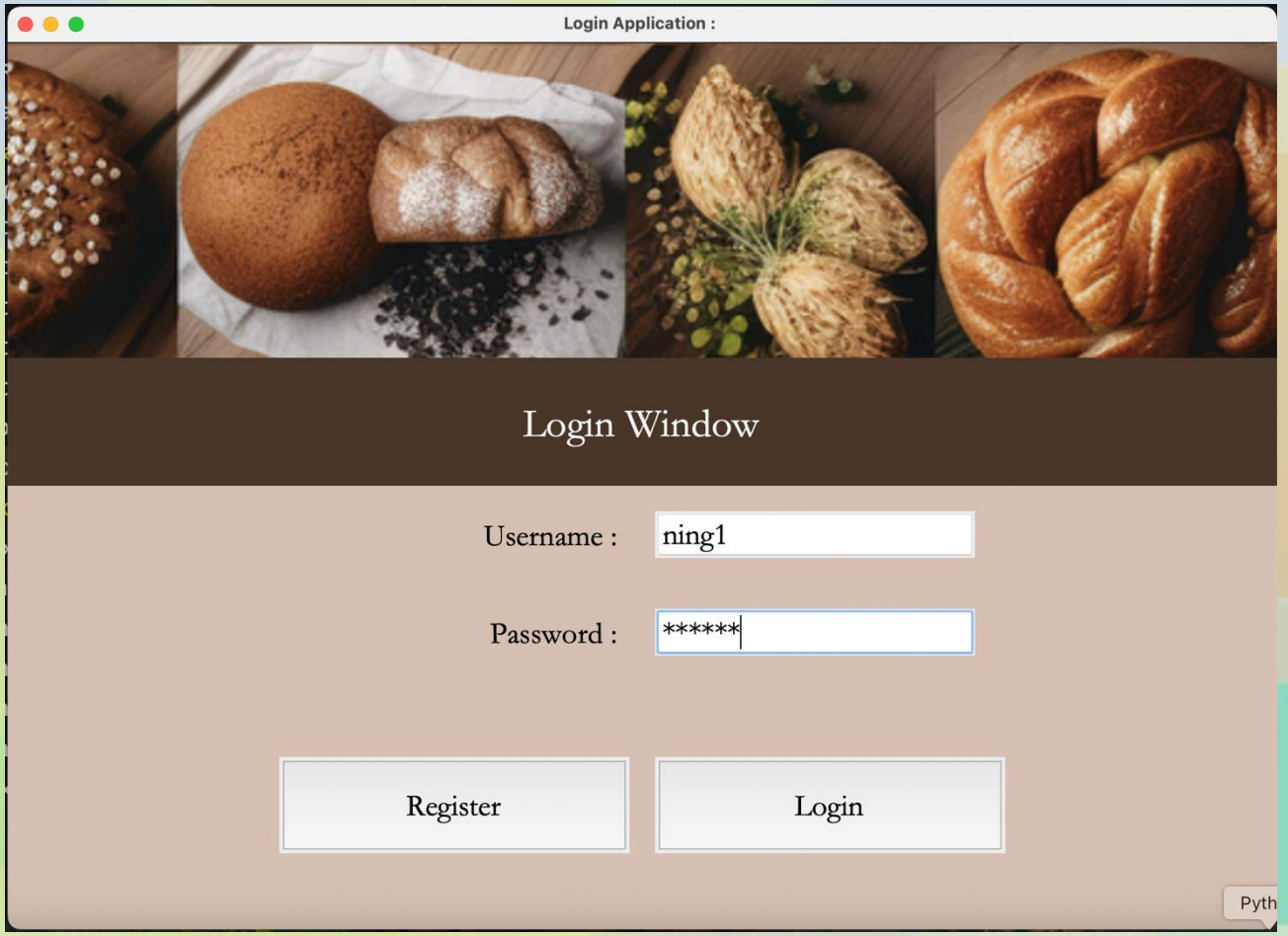
```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

WHERE CLAUSE USING AND/OR OPERATOR EXAMPLE

```
SELECT * FROM Customers
WHERE Country='Germany' AND City='Berlin';
```

```
SELECT * FROM Customers
WHERE City='Berlin' OR City=‘Greenway';
```

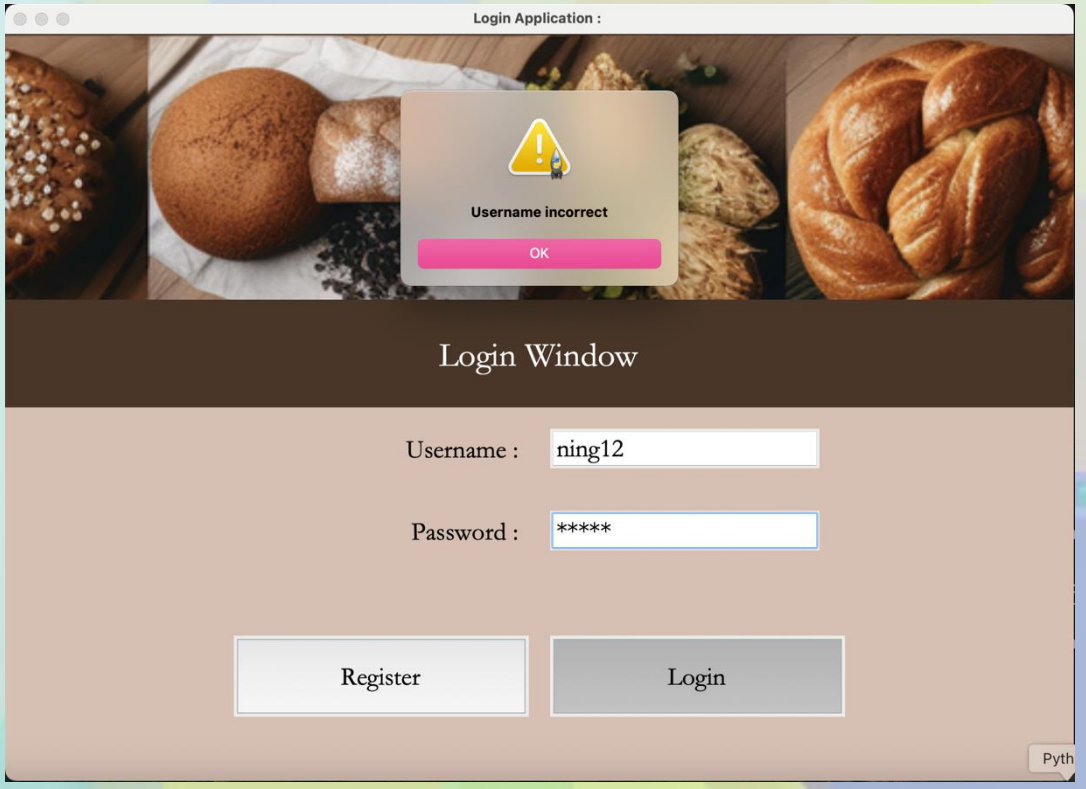
Class Activity of Week9

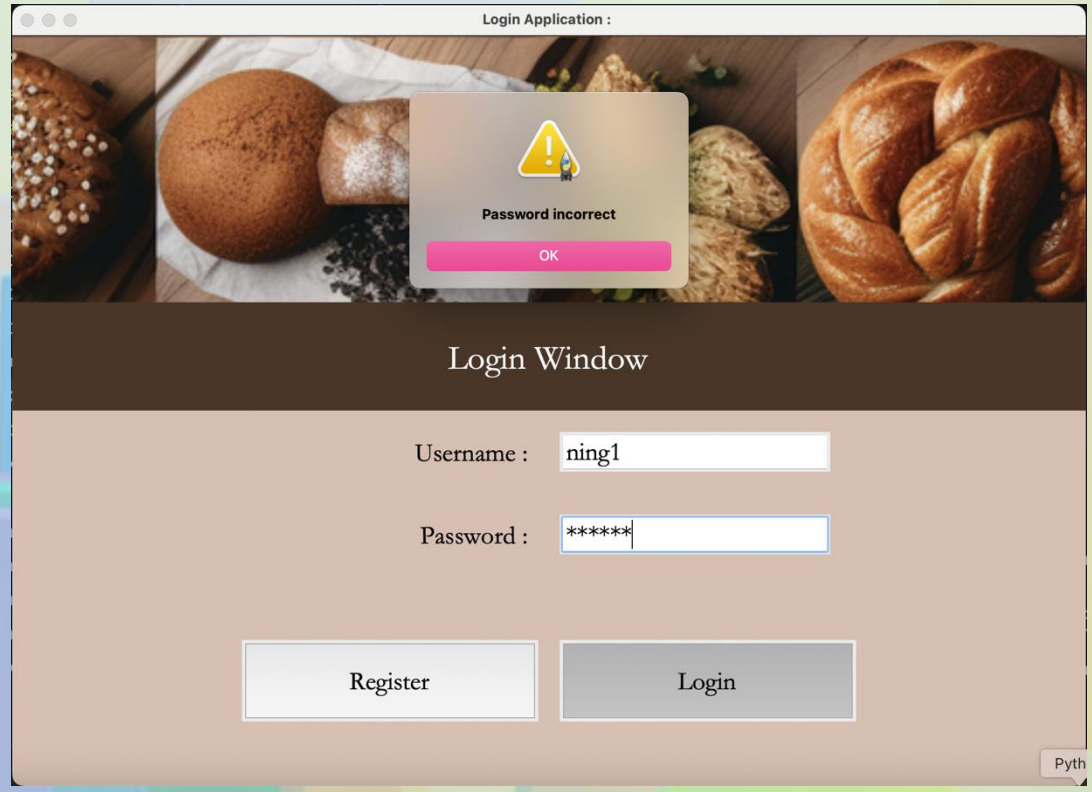


Login

Application

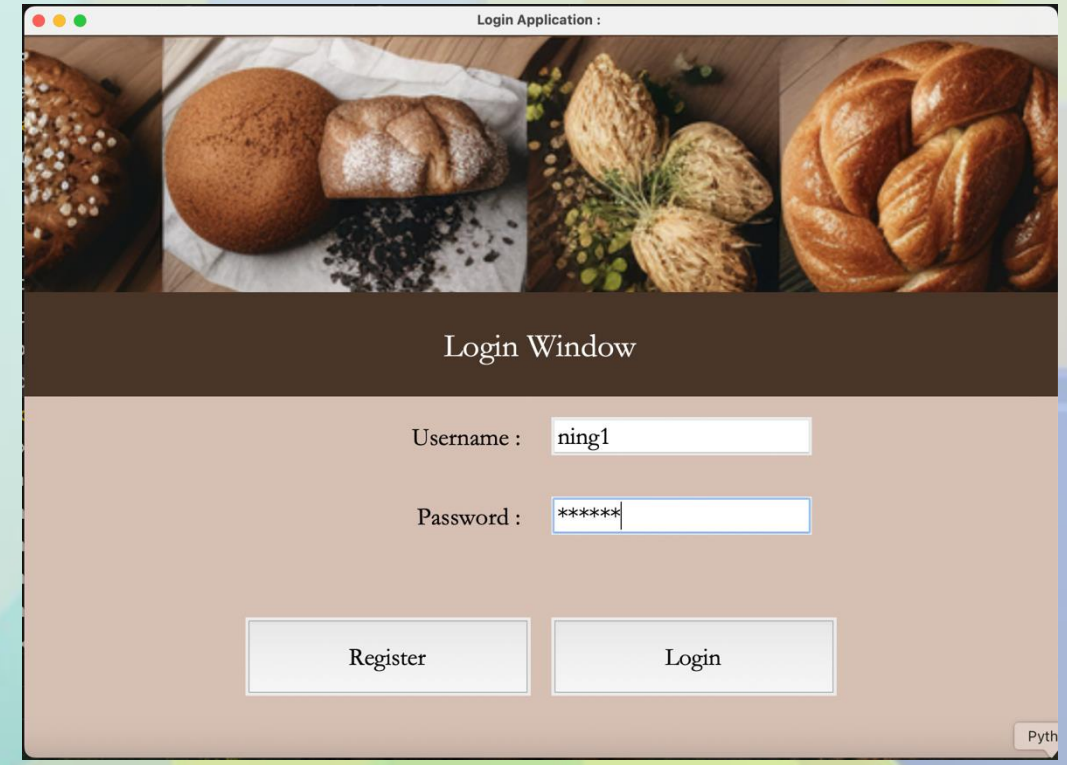
Login Application

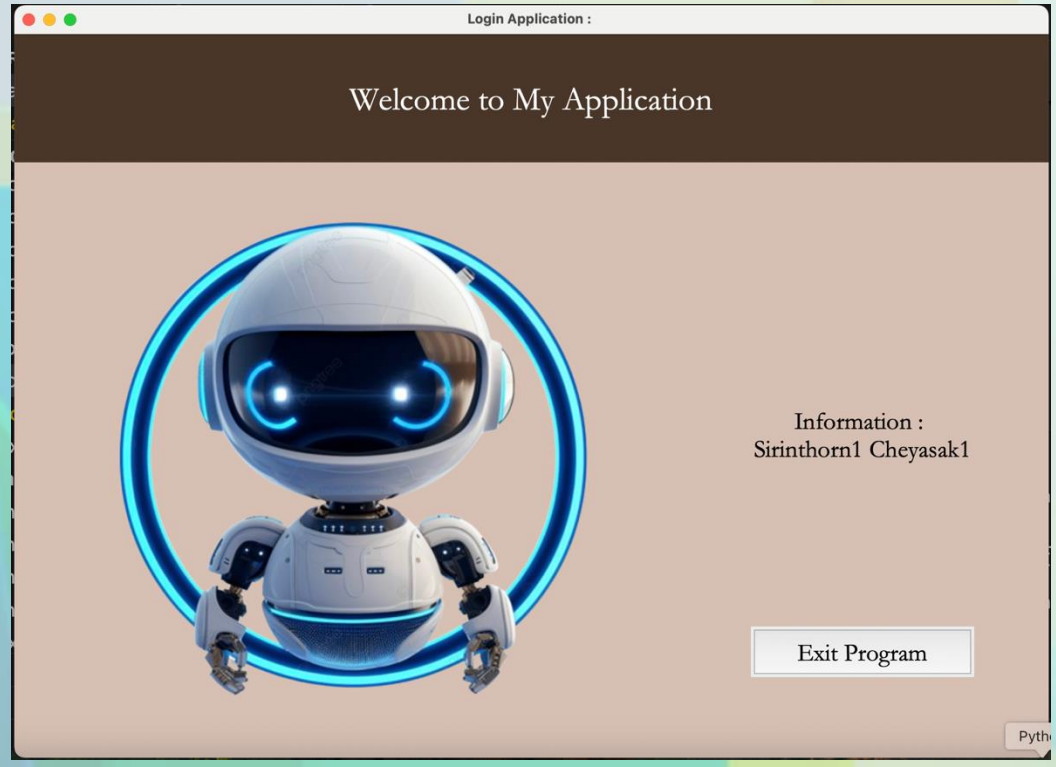




- WRITE A FUNCTION LOGINCLICK TO VERIFY BOTH OF USERNAME AND PASSWORD ARE EXISTS IN THE LOGIN TABLE

Login Application Output





See you next week