CS9053, Tuesday Section
April 29, 2020
Due May 6th, 2020, 11:55 PM
Prof. Dean Christakos

This is a problem set that requires both UI development and a database query. This isn't just about writing a SQL query. It's about providing an interface to work with the DB, getting the data back from the DB and displaying it in the interface.

I have included a sqlite file called assignment.db. This contains a table called "People". If you wish, you can learn use sqlite to access this file and explore the data. If you have a MacOS or Unix-based environment, you would do something like this:

```
Tyre:JDBCAssignment10SaturdaySection dean$ sqlite3 assignment.db
SQLite version 3.14.2 2016-09-12 18:50:49
Enter ".help" for usage hints.
sqlite> .tables
People
sqlite> .schema People
CREATE TABLE People (First text, last text, age int, city text, id INTEGER
PRIMARY KEY AUTOINCREMENT);
```

To manually insert an entry, you'd write:

```
INSERT INTO People (Last, First, age, city) VALUES ('Roy', 'Debajit', 50,
'Cambridge');
```

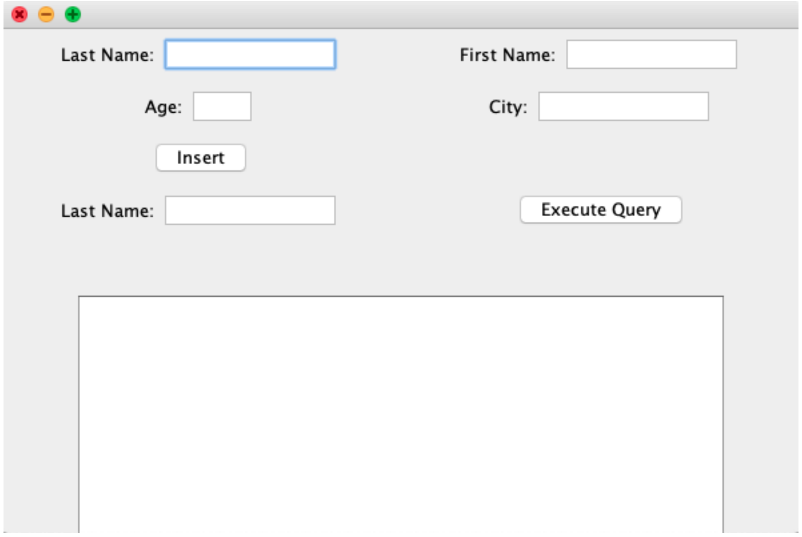
Notice that the id is automatically generated and incremented. You can see the results by executing:

```
SELECT * FROM People;
```

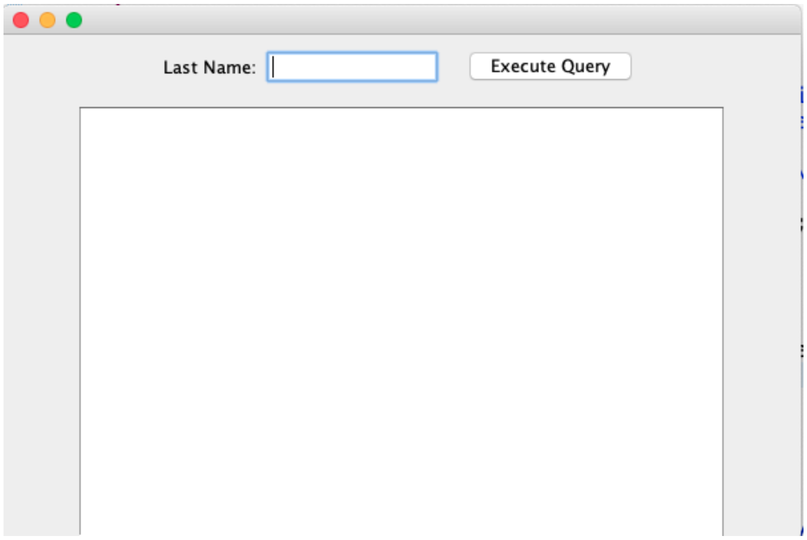
```
sqlite> select * from People;
Debajit|Roy|50|Cambridge|5
```

The ID column may be different depending on how many entries have been previously added.

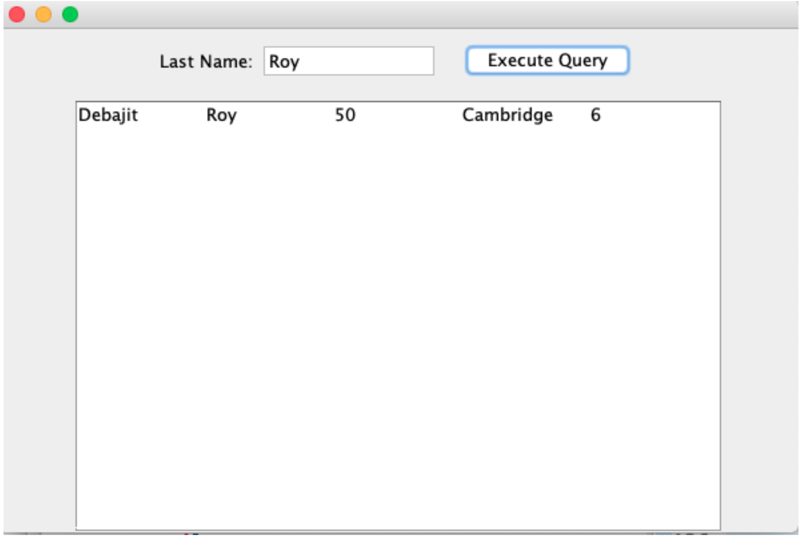
Your assignment is to create the following DB interface, using the skeleton code in `JDBCInterface.java`:



Currently all you have is this, so you’ll have to make some more components:



Current what happens here is that you can type a last name in to the text field, and then click “Execute Query,” which will show the results of all the rows, filtered for the Last Name:

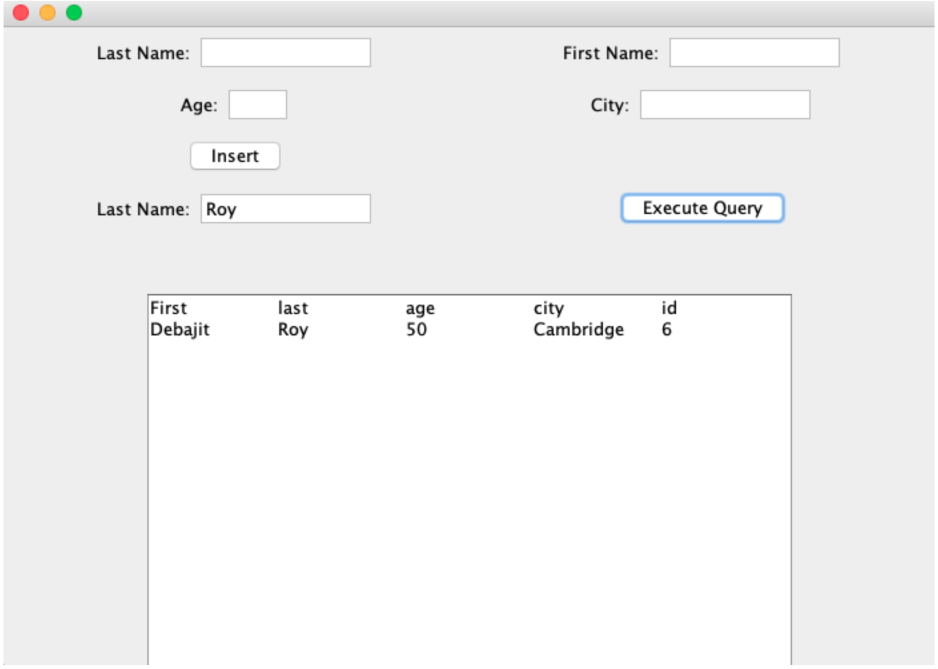


Last Name: Roy

```
Debajit         Roy             50              Cambridge       6
```

Your job is to duplicate the interface above to support inserting Columns and add a few features:

(a) Execute Query should return the column names above the rows. Get the data from ResultSetMetaData for this:

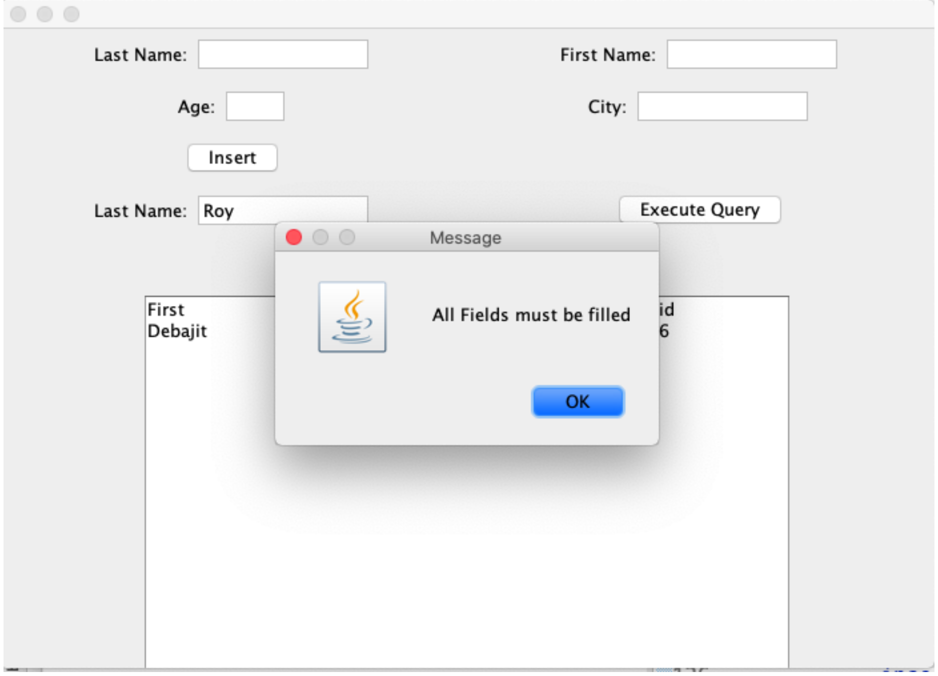


Last Name: Roy

```
First           last            age             city            id
Debajit         Roy             50              Cambridge       6
```

(b) If there is no text in the “Last Name” query field, ALL rows should be returned

(c) If you fill out the Last Name, First Name, Age, and City fields for Insert, then a new row should be inserted. You should use a Prepared Statement for this. This means creating an ActionListener tied to the Insert Button. If any of the fields are blank, you should raise an alert using `JOptionPane.`*`showMessageDialog`* like so:



Last Name: Roy

All Fields must be filled

First Debajit

id 6

(d) When you insert data, all of the insert fields should be erased after you press the button.

(e) You should be able to insert a few rows using the UI and then query all of the results

Notes and Hints:

You need to use a PreparedStatement to get full credit. There is already an example of a PreparedStatement for the query for last names. Do a similar thing for inserts.

To query all of the rows, you will have to create another query that doesn’t filter by last name and use that query instead of the one in queryStmtLastName.

Read the Java API to understand the functions available from ResultSetMetaData.

There is some sample code in JDBCExample.java that you can work with that shows some simple examples with the People table.