SQLite

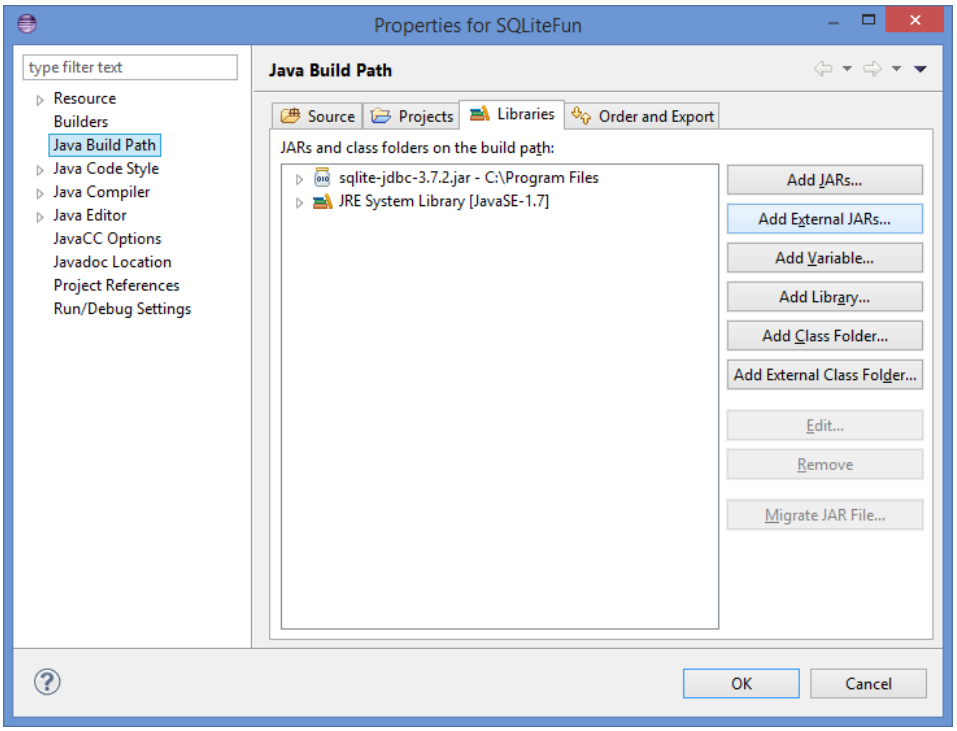
https://bitbucket.org/xerial/sqlite-jdbc/downloads

Go to this website and download the sqlite jar file.

Downloads | Tags | Branches

Name	Size	Uploaded by	Downloads	Date
Download repository	107.8 MB			
sqlite-jdbc4-3.8.2-SNAPSHOT.jar	3.6 MB	gbatumbya	123848	2014-01-05
sqlite-jdbc-3.7.15-M1.jar	3.5 MB	xerial	75728	2013-08-19
sqlite-jdbc-3.7.15-SNAPSHOT-2.jar	3.5 MB	gbatumbya	9243	2013-03-24
sqlite-jdbc-3.7.2-javadoc.jar	132.7 KB	xerial	12179	2013-01-23
sqlite-jdbc-3.7.2.jar	3.1 MB	xerial	76644	2013-01-23
sqlite-jdbc-3.7.15-SNAPSHOT-javadoc.jar	128.2 KB	gbatumbya	2305	2012-12-16

Next, go ahead and startup a new Java project in Eclipse. Right-Click on the project and select Build Path→ Configure Build Path. Under the Libraries tab, Select Add External JARs and find the Sqlite jar file.



Now that you have SQLite connected to your project, you can test it out using the following snippet. If everything is set up properly, the console should have been written to.

```
import java.sql.*;

public class SQLiteJDBC
{
  public static void main( String args[] )
  {
    Connection c = null;
    try {
      Class.forName("org.sqlite.JDBC");
      c = DriverManager.getConnection("jdbc:sqlite:test.db");
    } catch ( Exception e ) {
      System.err.println( e.getClass().getName() + ": " + e.getMessage() );
      System.exit(0);
    }
    System.out.println("Opened database successfully");
  }
}
```

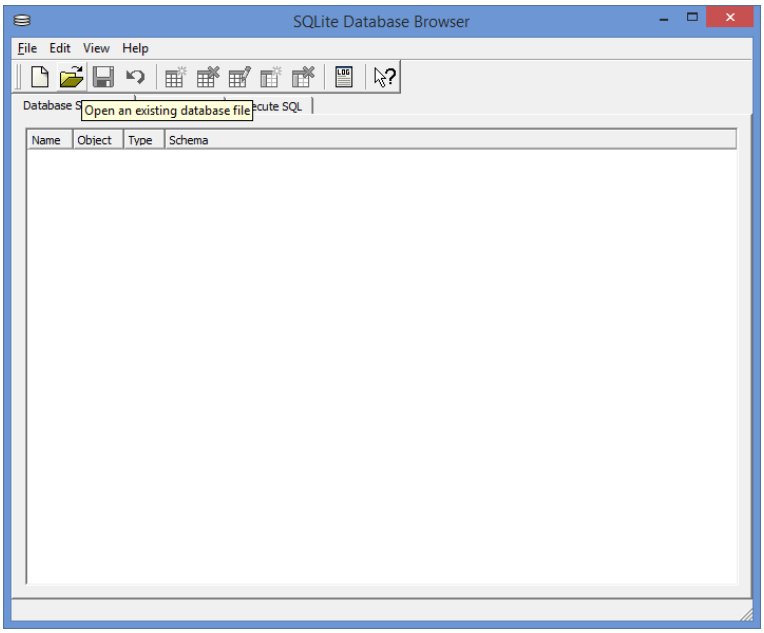
This is an example of adding a table.

```
      stmt = c.createStatement();
      String sql = "CREATE TABLE COMPANY " +
                   "(ID INT PRIMARY KEY     NOT NULL," +
                   " NAME           TEXT    NOT NULL, " +
                   " AGE            INT     NOT NULL, " +
                   " ADDRESS        CHAR(50), " +
                   " SALARY         REAL)";
      stmt.executeUpdate(sql);
      stmt.close();
```

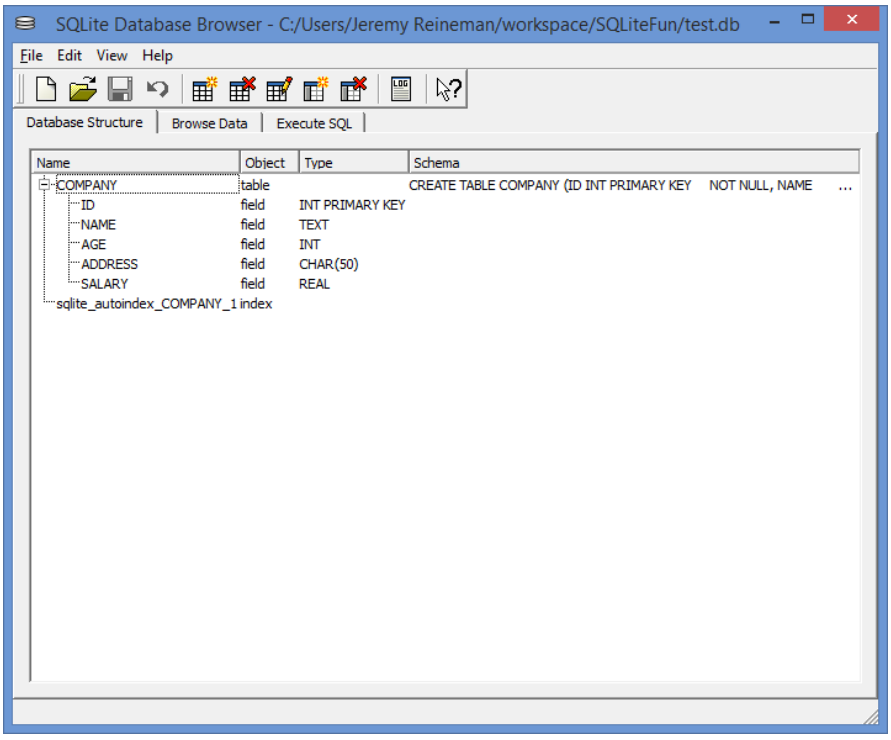
Next adding some tuples to the table.

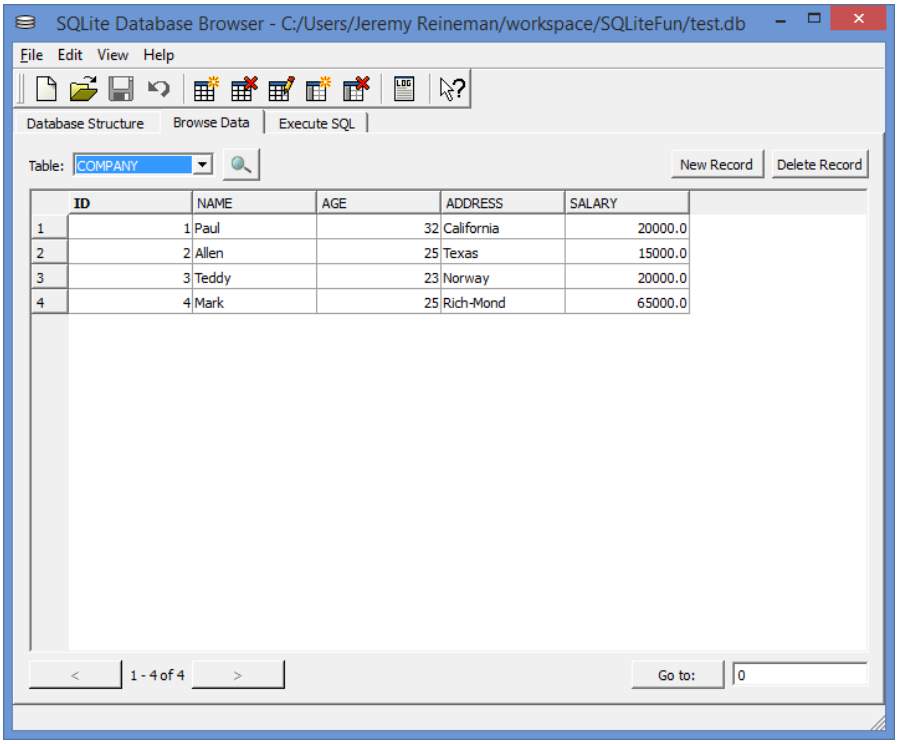
```
stmt = c.createStatement();
String sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
             "VALUES (1, 'Paul', 32, 'California', 20000.00 );";
stmt.executeUpdate(sql);

sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
      "VALUES (2, 'Allen', 25, 'Texas', 15000.00 );";
stmt.executeUpdate(sql);

sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
      "VALUES (3, 'Teddy', 23, 'Norway', 20000.00 );";
stmt.executeUpdate(sql);

sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
      "VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00 );";
stmt.executeUpdate(sql);

stmt.close();
c.commit();
c.close();
```

Now that there is data in the database, it would be nice to view it. Head to http://sourceforge.net/projects/sqlitebrowser/ and download this lightweight database browser. Once you have the application open, you can select Open an existing database file to view the newly made database which should be in your project folder.



Now you can see the newly added tuples to your database.





You can also modify the database from this application as well. For more commands and other great tutorials, go to http://www.tutorialspoint.com/sqlite/sqlite_java.htm which is where the previous code originated from.