**Advanced Programming - Student RESTful Web Service Coursework**

**STEP 1**

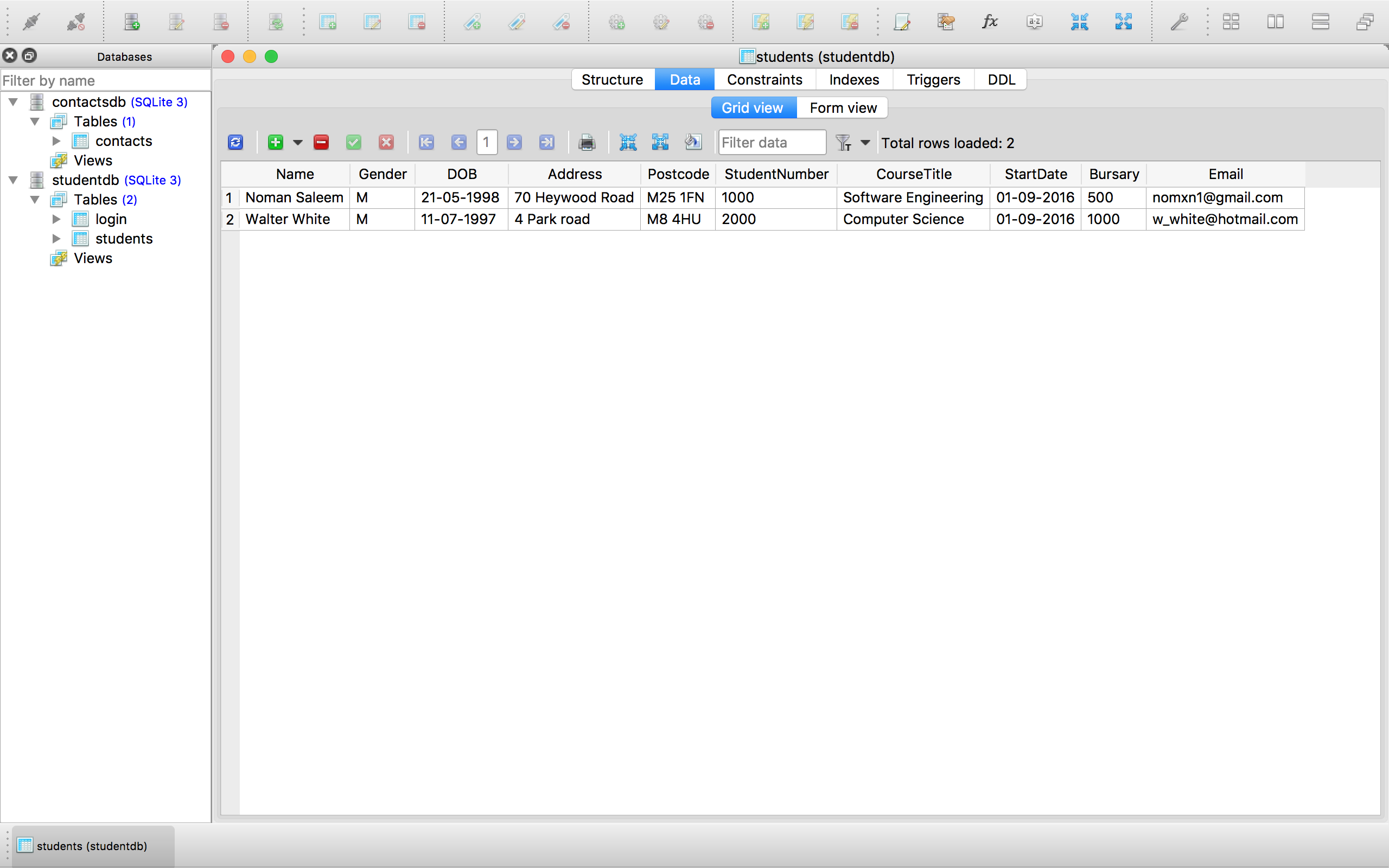


Figure - All the records in the database.

A database has been created called studentdb.sqlite which includes a table called students. The ‘students’ database includes two student records as shown in ***FIGURE 1***. The primary key in the table is the ‘StudentNumber’ field giving each student a unique ID number.

**STEP 2**

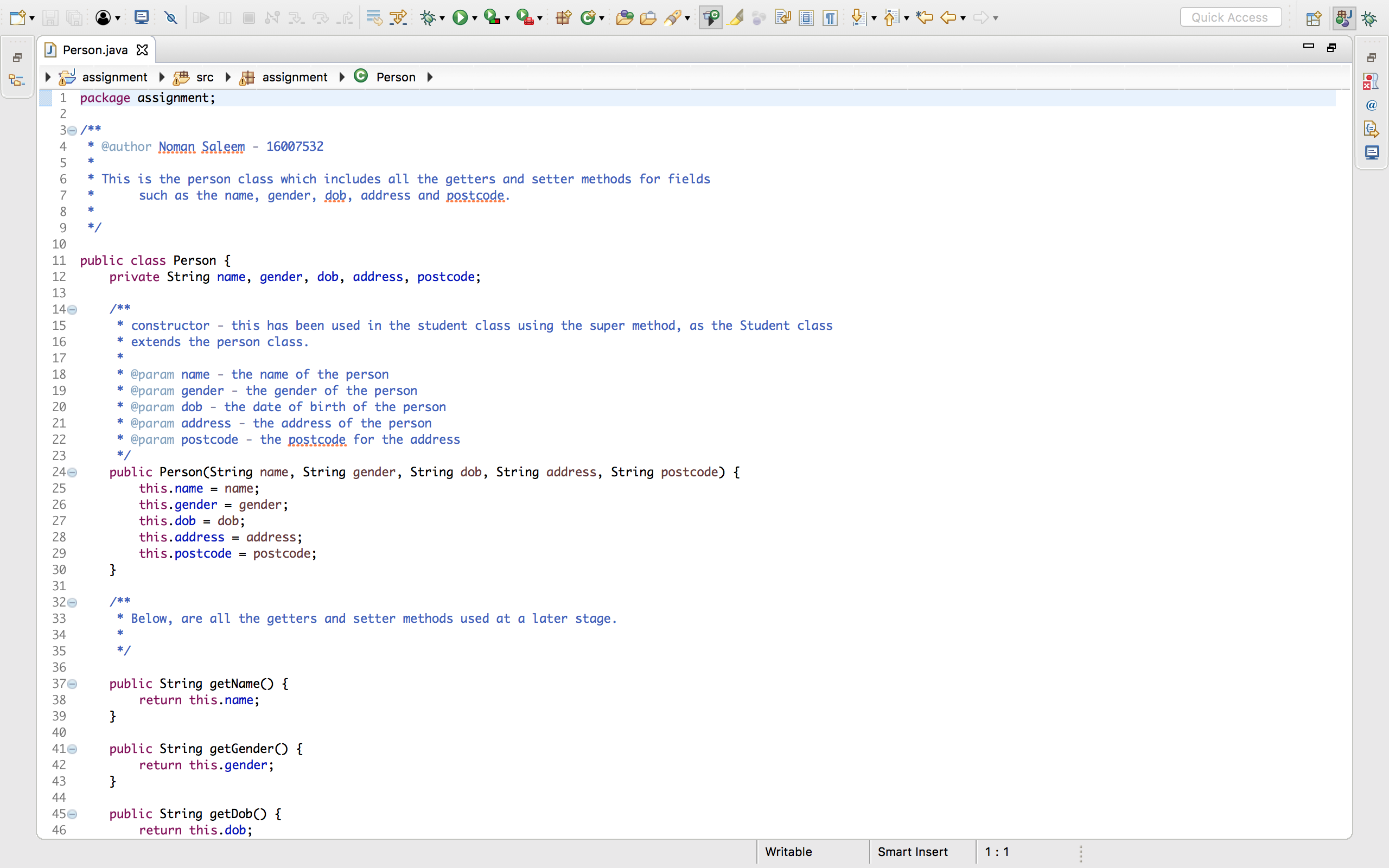
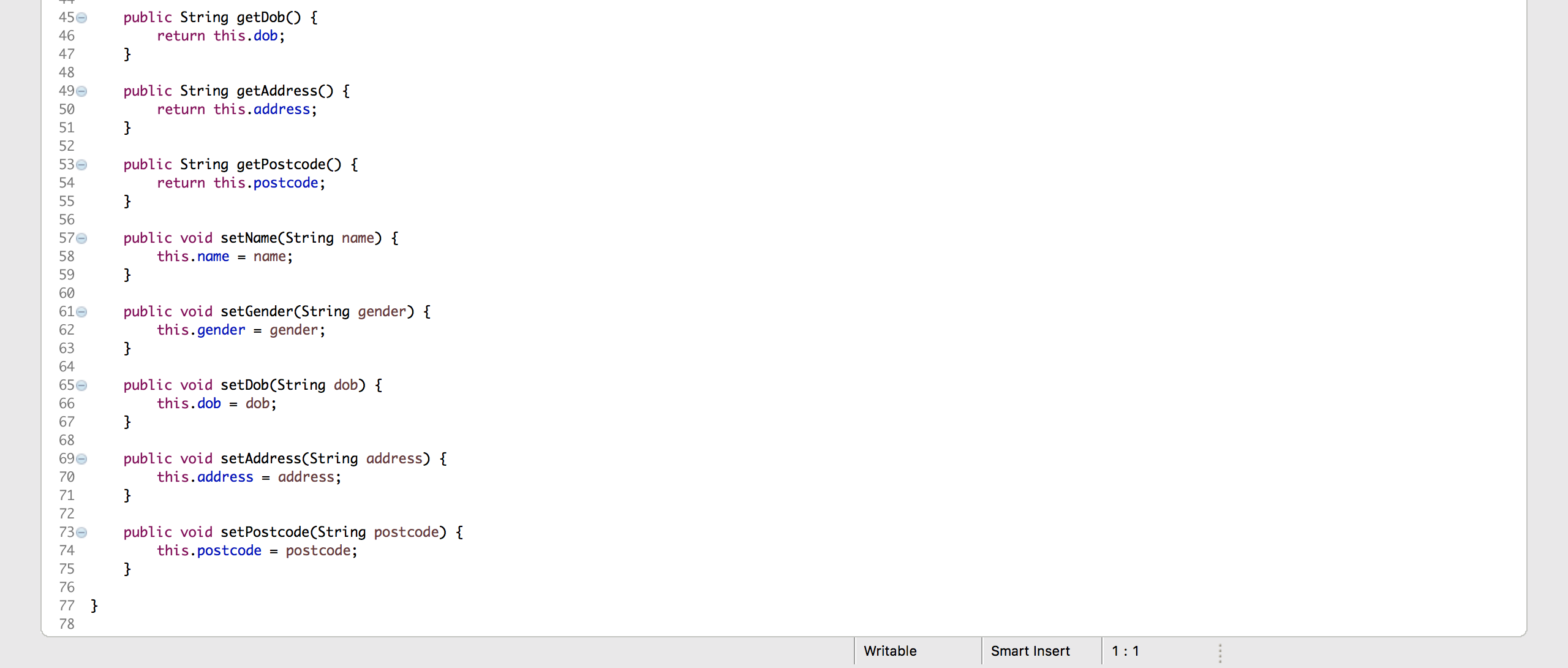


Figure - Code for the person class.

Figure 2.1 - Code for the person class.

The first class in this Java program is the Person class, shown in **FIGURE 2** and **2.1** which includes variables such as name, date of birth, gender etc. to match the fields in the database.

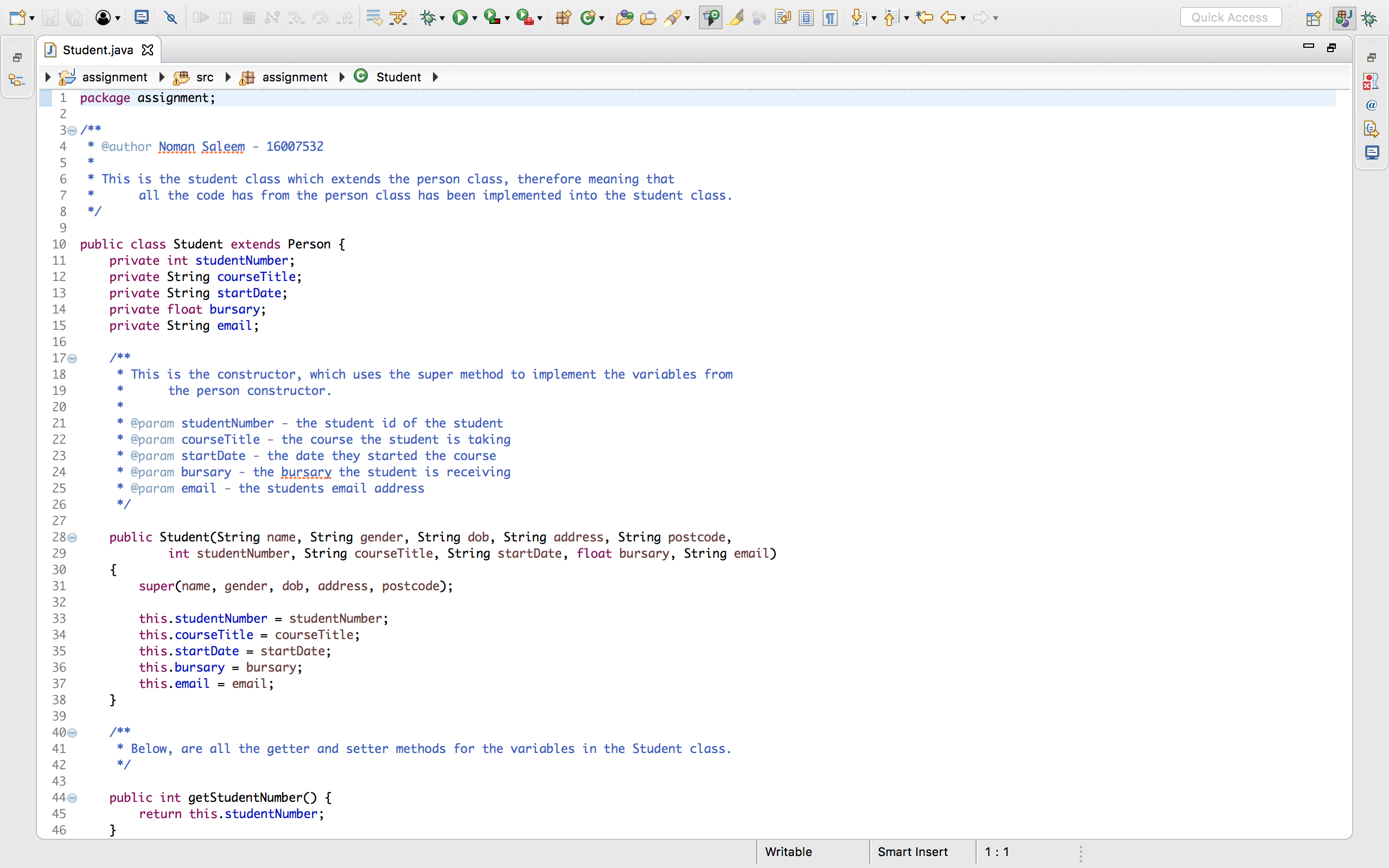


Figure 3 - code for student class.



Figure 3.1 - code for the student class.

The next class created was the Student class which extends the Person class in order to inherit the getter and setter methods from the Person class. A super method was used in the constructor to ensure the extra variables in the Student class were connected with the variables in the Person class. E.g. the StudentNumber etc. See **FIGURES 3** and **3.1**.

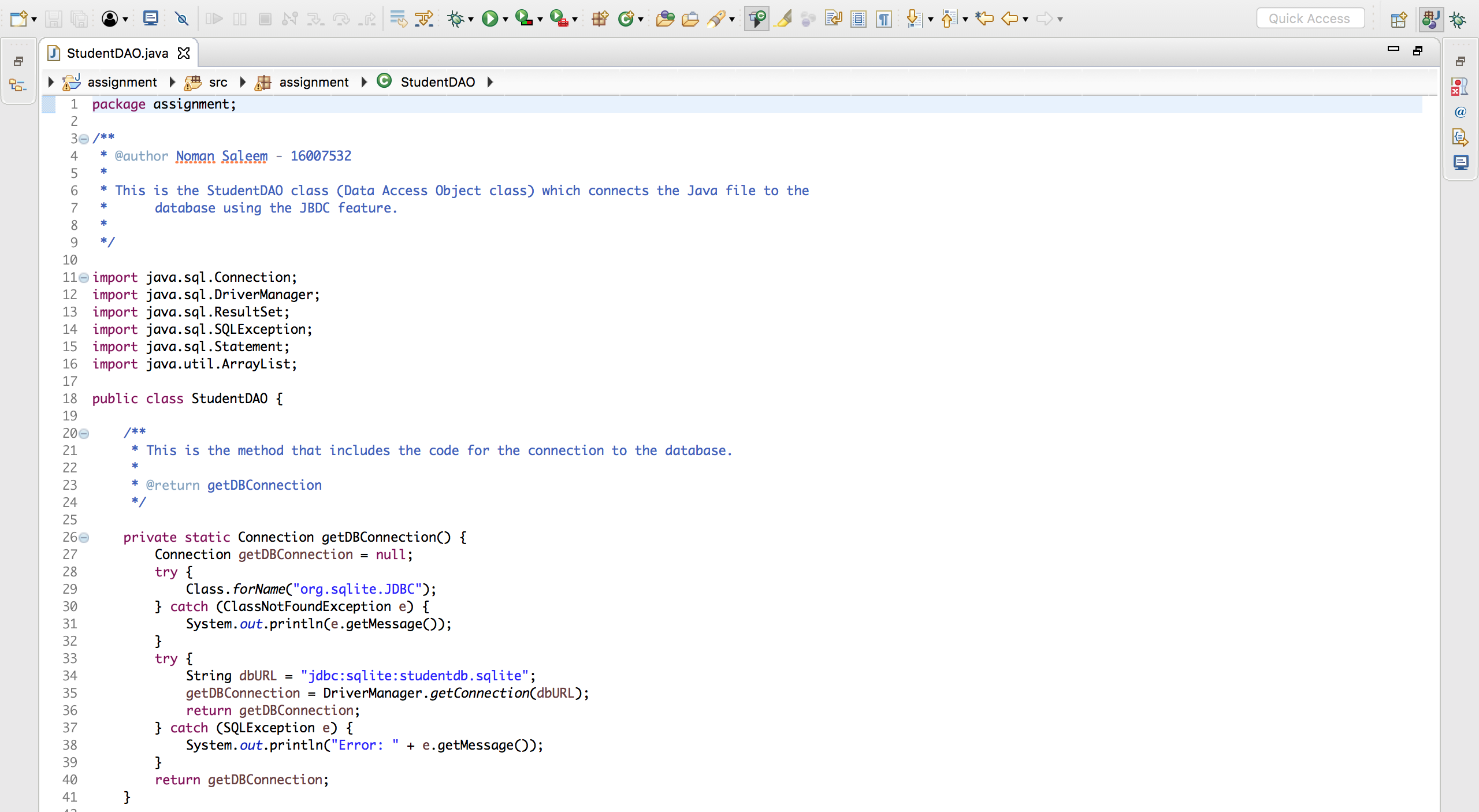


Figure 4 - Code for the student dao class.

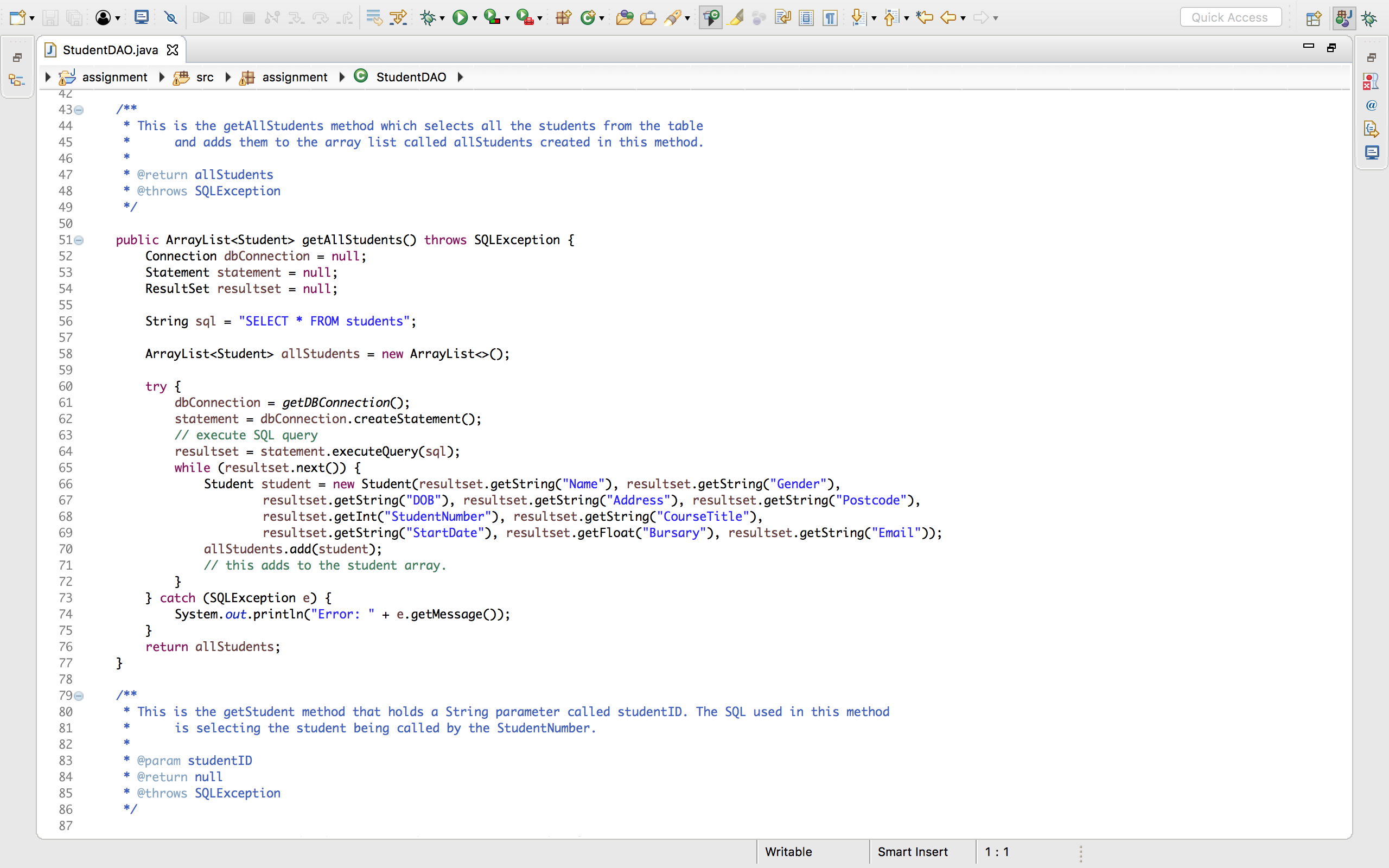


Figure 4.1 - code for the student dao class.

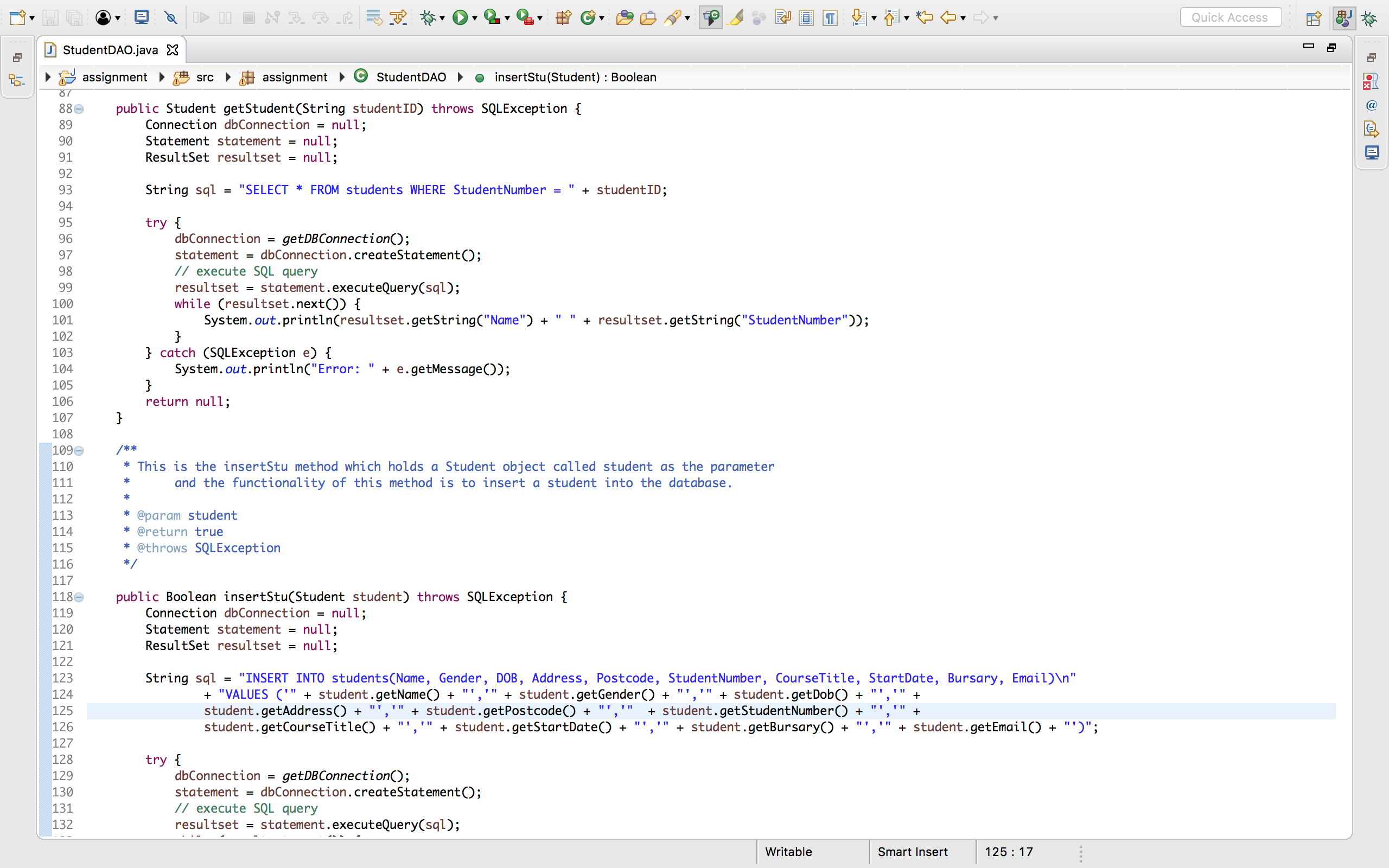


Figure 4.2 - CODE FOR THE STUDENT DAO CLASS.

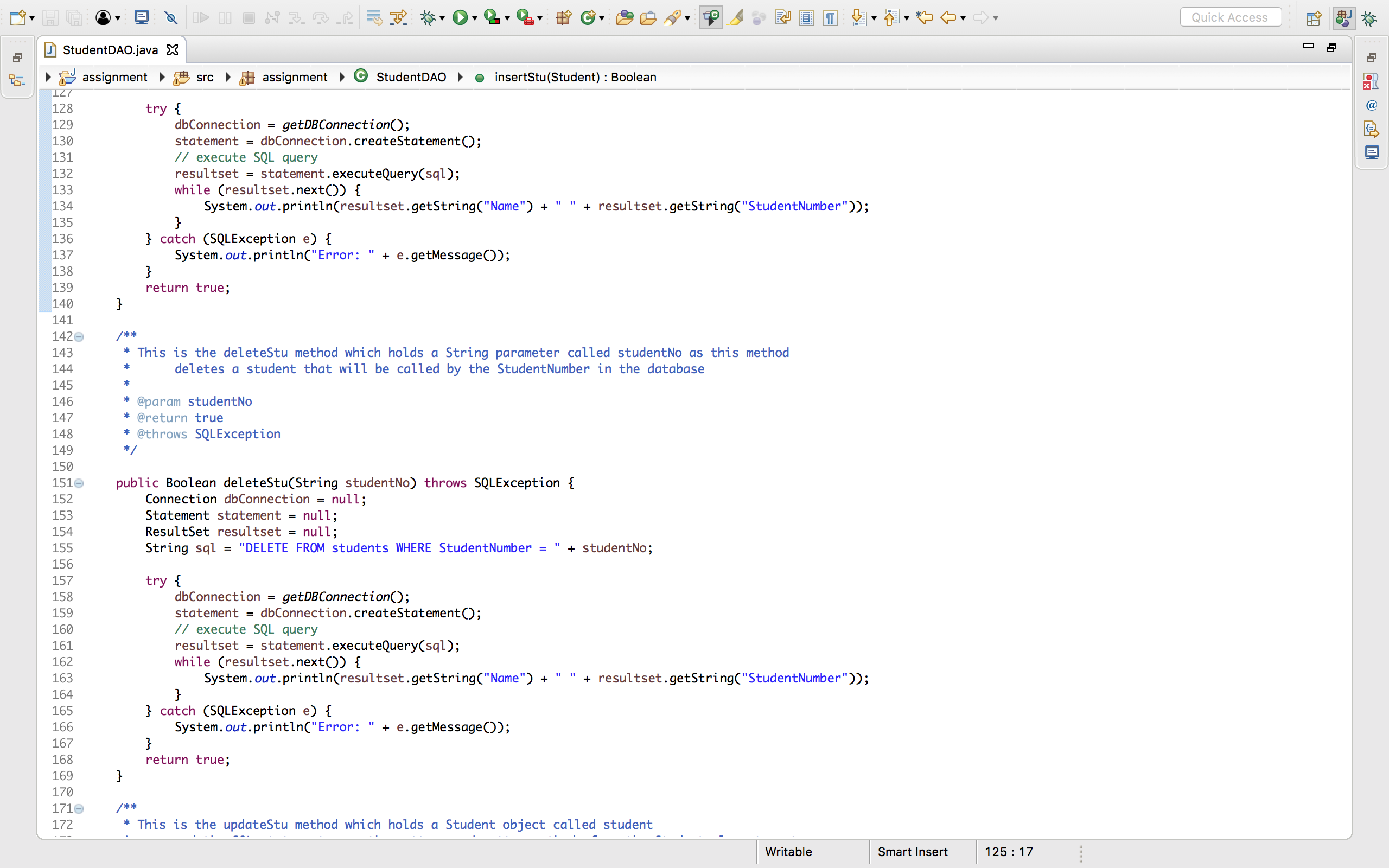


Figure 4.3 - CODE FOR THE STUDENT DAO CLASS.

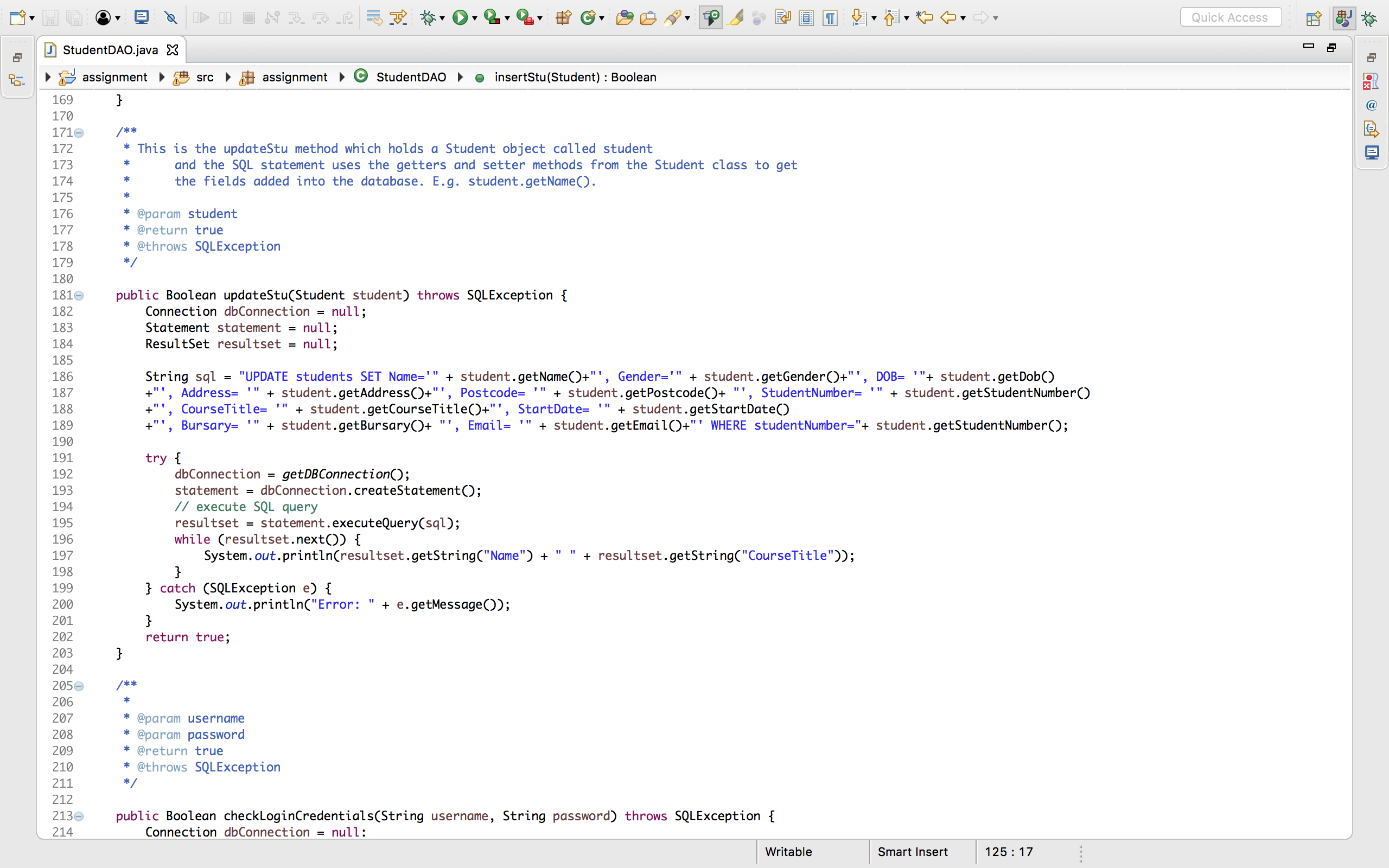


Figure 4.4 - CODE FOR THE STUDENT DAO CLASS.



Figure 4.5 - CODE FOR THE STUDENT DAO CLASS.

**FIGURE 4** to **4.5** show screenshots of the StudentDAO class which was responsible for the database connection being created, which is highlighted in FIGURE 4, more specifically, the getDBConnecton method. The remaining screenshots show all the methods needed for the CRUD operation feature. For example, getAllStudents, insertStu, and deleteStu etc.

**STEP 3**

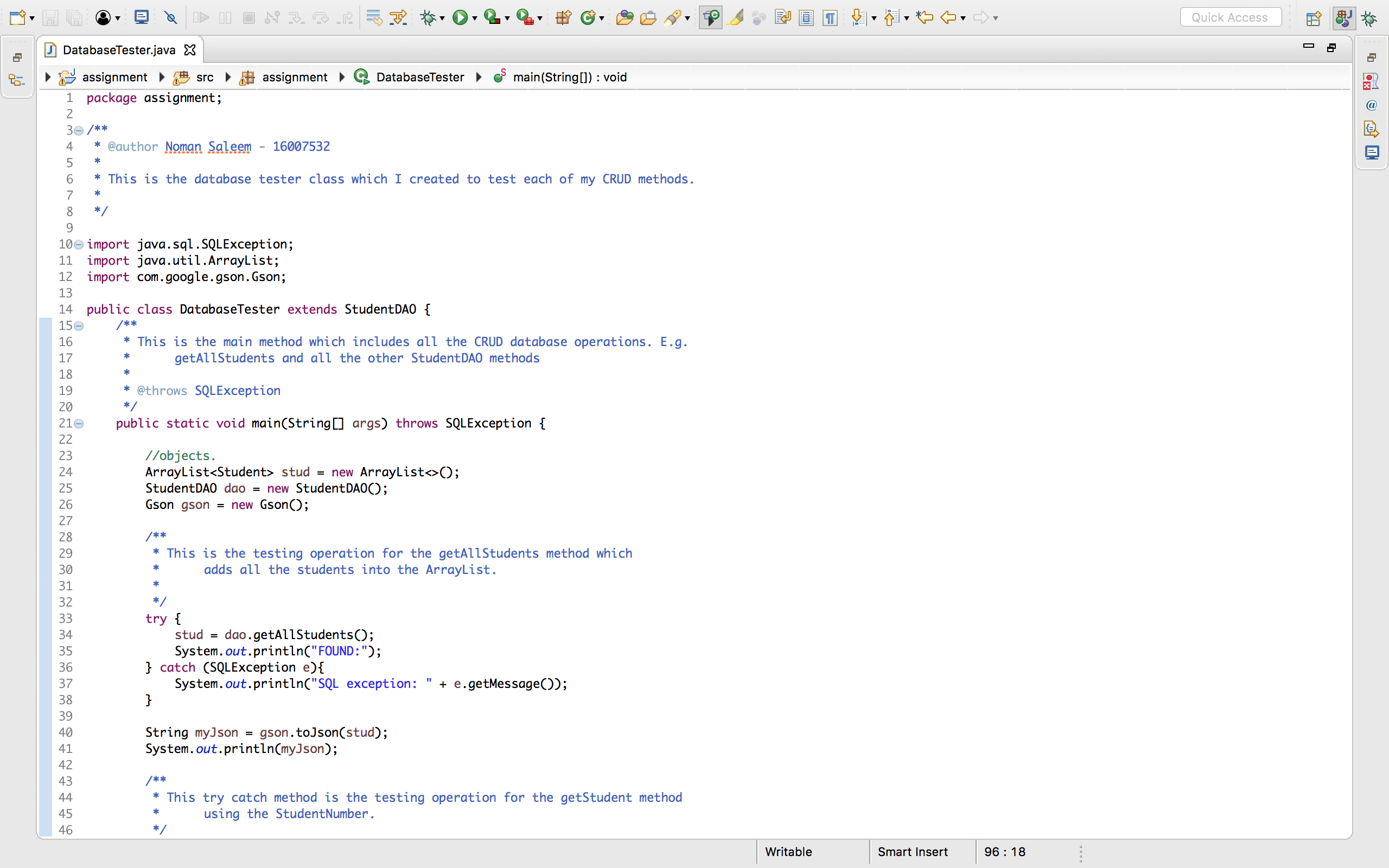
****

Figure 5 - Code for the database tester class.

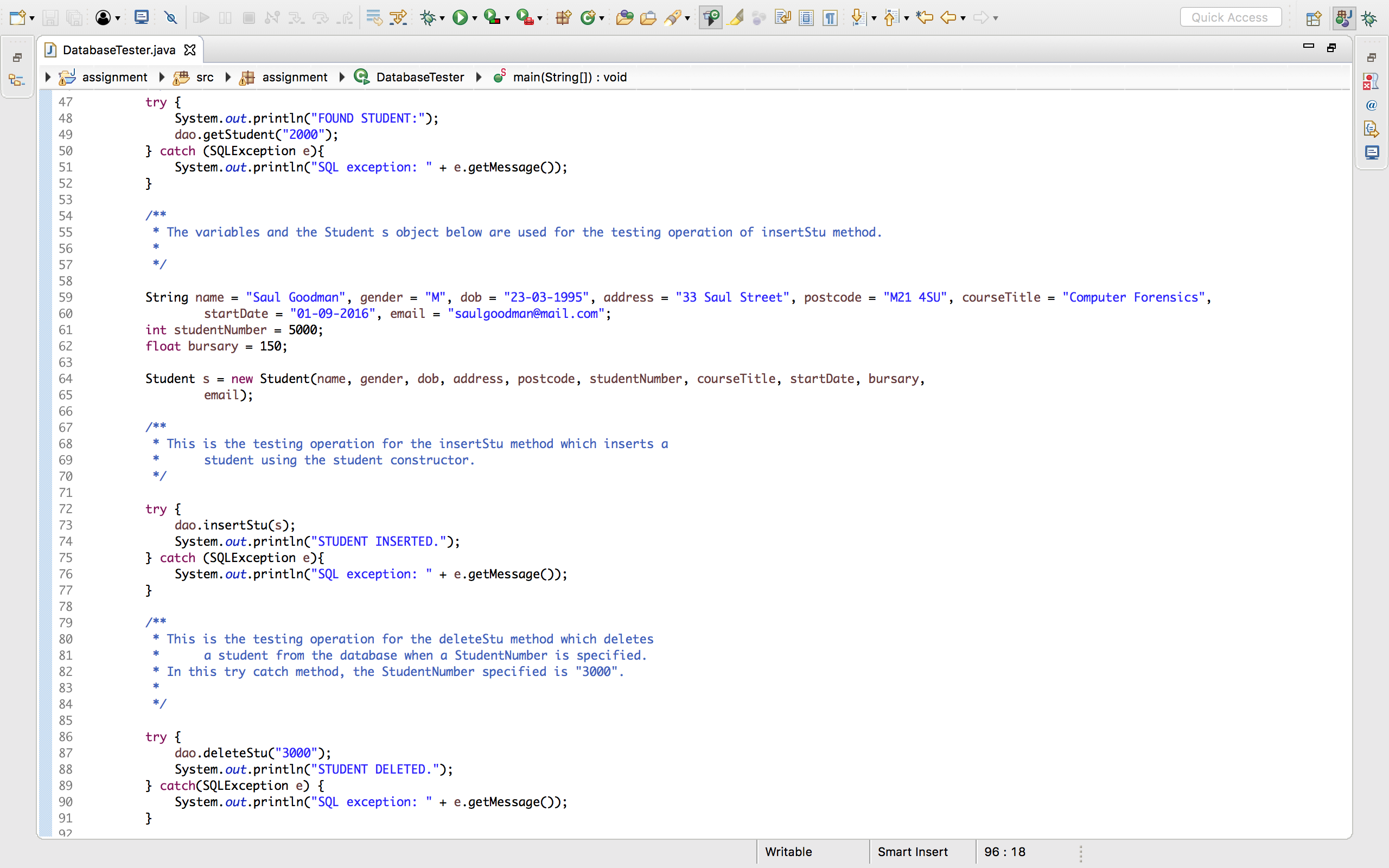
****

Figure 5.1 - code for the database tester class.

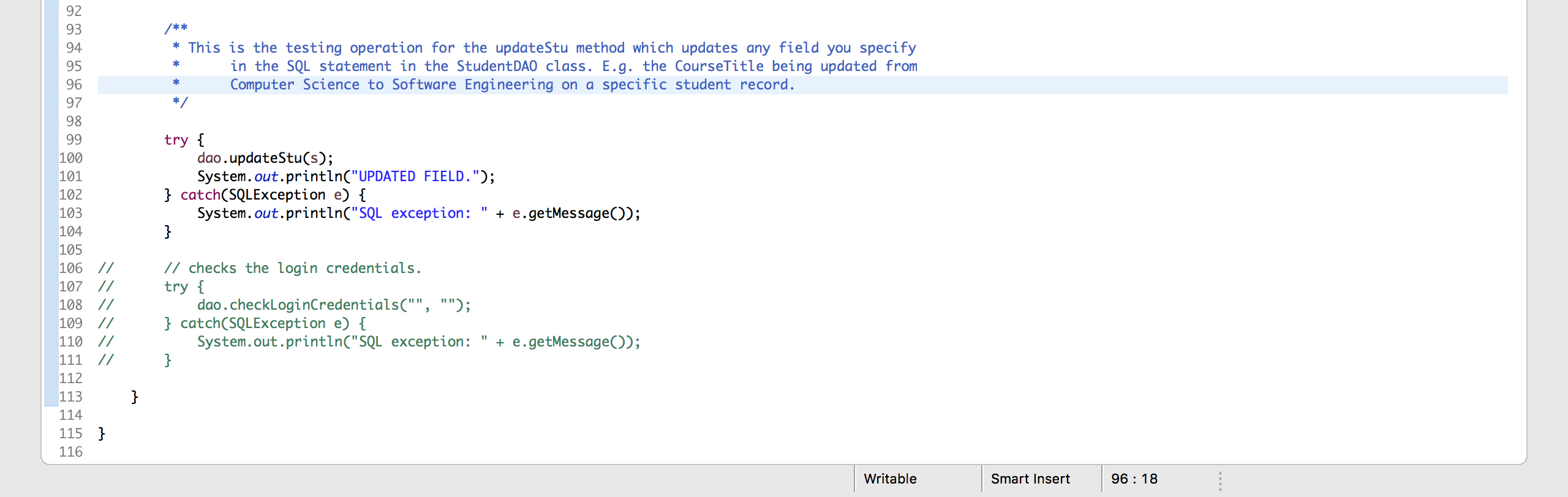
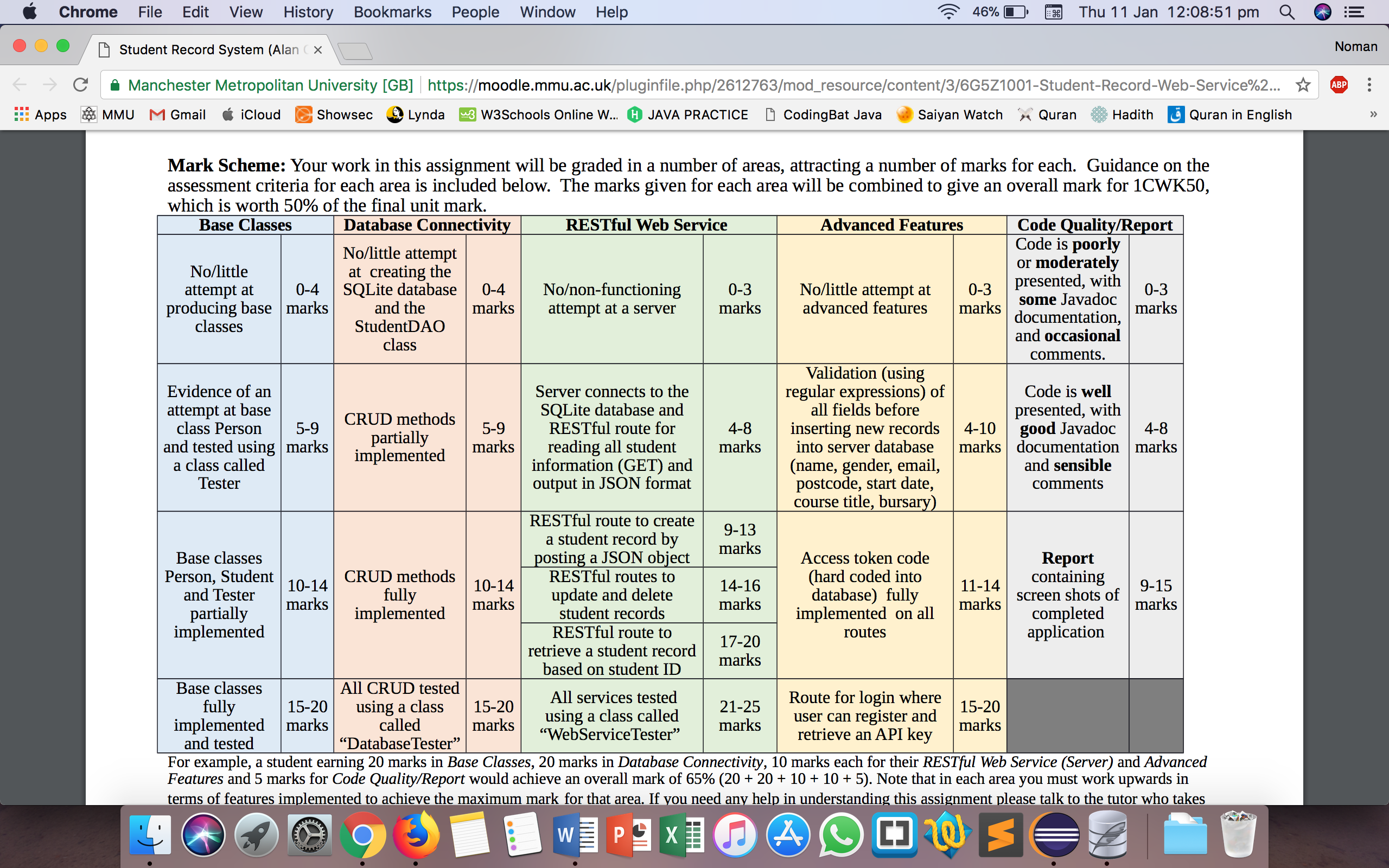
****

Figure 5.2 - code for the database tester class.

After all the base classes were successfully implemented, to test these CRUD operations, a DatabaseTester class was created to test these operations. **FIGURE 5** to **5.2** show the different testing operations produced to successfully test each of the CRUD operations. For example, in **FIGURE 5.1** for the Create testing, a testing operation for the insertStu method has been written, which inserts a student in the database.

**STEP 4**

****

* Base Classes – All the base classes have been successfully implemented and tested. See **STEP 2**.
* Database Connectivity – All CRUD methods tested using a class called “DatabaseTester”. See **STEP 3**.
* RESTful Web Service – Completed up to RESTful routes to update and delete student records.