Databases

Exercise 7
JDBC

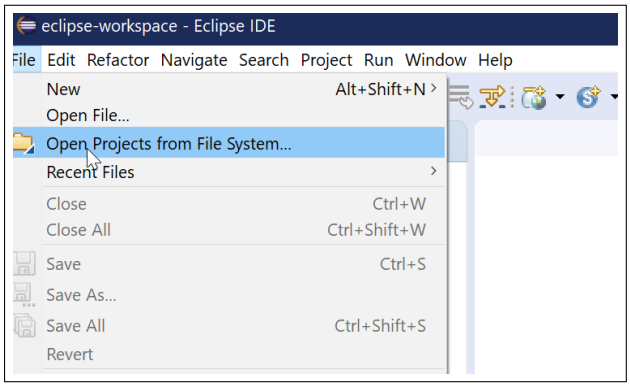
Assistants:
Maxim Van de Wynckel (Maxim.Van.de.Wynckel@vub.be)
Ruben Opdebeeck (Ruben.Denzel.Opdebeeck@vub.be)

Requirements

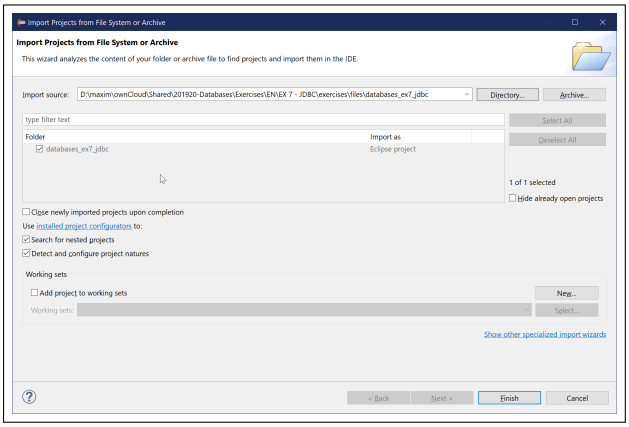
For this exercise we will make use of JDBC (Java Database Connectivity) to communicate with an SQLite database. In order to solve the following questions you will need to make the following preparations:

1. Install Eclipse and JDK 8 or higher if not on the VUB computers
   (a) Download and install JDK (Java Development Kit) 8 or higher from oracle.com
   (b) Download the latest Eclipse IDE from eclipse.org
   (c) Start the installation for Eclipse. You will be asked to specify the type of projects you need Eclipse for. It is advised to select the normal "Eclipse for Java Developers" option.

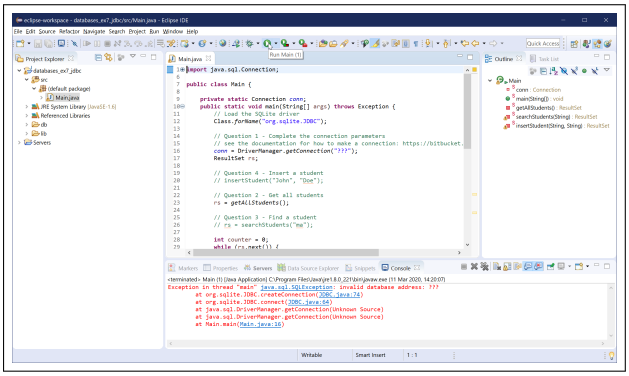
2. Download and extract required files from Cnavas
3. Import the project in Eclipse
   (a) Go to File > Open Projects from File System...



   (b) Browse to the extracted directory with the Eclipse Project



(c) Double click the "Main" file under "src" to view the code and run it by pressing the play button. You will see an error, but this is normal.



4. Create the SQLite database "students.db" with SQLite Studio as described in the tutorial of the previous exercise session. Alternatively you can download the SQLite database from Canvas.

1 Question 1

As described in the lecture slides (Lecture 6, Advanced SQL, Slide 30) we have to create a JDBC connection before we can execute queries. At the bottom of the slide you can see that the JDBC connection URL is different for each JDBC driver (MySQL, SQLite, PostgreSQL, ...). Because we are using SQLite the connection URL will be different from the example given on the slides. If you try to run the code without modifying the URL you will get the following error:

"invalid database address: ???"

For this first question you will have to construct thee correct JDBC connection URL in order to connect with the previously constructed database file "students.db". Put the file in the "db" subfolder of the project and create the correct connection URL. Once the URL is correct you will see that the error changes to:

"[SQLITE_ERROR] SQL error or missing database (near "?": syntax error)"

This is normal and we will fix this in the next question

Hint: Check the official SQLite JDBC driver documentation for more information:
https://bitbucket.org/xerial/sqlite-jdbc/wiki/Usage

2 Question 2

As mentioned earlier you should now get the following error:

"[SQLITE_ERROR] SQL error or missing database (near "?": syntax error)"

This is because the function *getAllStudents()* has not been implemented completely. Normally this function will execute a query to retrieve a list of all students (from the Student table). However, the query has not been filled in yet and is shown as "???" in the code. Complete the *getAllStudents()* function by writing a simple SQL query that gives us a list of all students. You just need to replace the "???" with your query, no further adjustments are needed. The *Main()* function will automatically iterate over the results and will display *StudentFirstname* and *StudentSecondname* in the console.

3 Question 3

For the next step we will implement some simple search functionality for our database. We would like to find students based on their name or parts of their name. For instance, if we search for "ma" we would like to get two results: Emanuel Roels and Alex Maes. Some example code is given in the *searchStudents()* function. Write an SQL query that performs a search for a given string. The

searchStudents() function has one parameter ("name") which is a string containing the (partial) name of the students that we want to look for. In other words, write an SQL query that returns students where either *StudentFirstname* or *StudentSecondname* contains the string name. Do not forget to remove the comment delimiter before the line:

```
rs = searchStudents("ma");
```

when testing your query.

4 Question 4

For this question we will write some code to add new students to our database. To achihve this we will make use of so-called "prepared statements". Use Google to find some examples and documentation on prepared statements in JDBC. Just like the previous question toy do not need to adjust any code and just need to replace the "???" with an SQL query. To test your query, make sure that *insertStudent()* and *getAllStudents()* are called and make a comment out of the *searchStudents()* call from the previous question.

5 Question 5

Finally, use the example code from the previous questions to build a function that displays each *StudentGrade* together with the amount of students in that grade. Note that you will also have to modify the code that iterates and prints the result set.