Database Java Project

Stancu Tudor-Andrei, 1533e

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

For this project I decided to have database with 4 fields that describe a person’s first name, last name, date of birth and an index for easier referencing. The project’s GUI part is made using JAVA Swing, the backend is made using JAVA and the database managing is done in Microsoft Access.

Graphical user interface

Description automatically generated

On the GUI part we have a few different components. A JScrollPane that displays the requested data from the database. Three jTextFields from where the data is taken and processed when one of the five jButtons are pressed. We also have two jCheckBoxes that do a certain task when checked.

The data from the database is stored in a table (Table2 in our case ) with 4 columns: ID, nume(string), prenume(string) and dataNastere(DateTime).

Table

Description automatically generated

\*the data presented in the table is for example purposes

For the functional part we have five different buttons:

* Add – adds a new row of data using the user introduced data
* Delete – deletes a row of data that the user selected
* Update – replaces the data from a selected row with the data introduced
* Search – displays all of the people with the name introduced in the “Nume” field
* Refresh – refreshes the table and displays all of the data present without any filters

We also have two check boxes:

* Sort by name – displays the whole table ordered in alphabetical order by the “Nume” field
* Is older than 18 – displays all the people that are older than 18

The link to the database is done using this string variable called “url” which leads us to the exact location in our computer where the actual database tables are stored. This variable is public and is instantiated a single time at the start of the application.

Each time we run the application this piece of code is ran:

Graphical user interface, text, application, email

Description automatically generated

This ensures that the text fields are empty and check boxes are unchecked. It also uses a void method called refresh() that displays the whole database.

Graphical user interface, text, application

Description automatically generated

In the first part of the method we are connecting to the database and also deleting everything that previously was displayed in the jTable. We run a query on the database that selects all fields following to redisplay everything in the UI. It also deselects the check boxes in case they are left selected by mistake.

Graphical user interface, text, application

Description automatically generated

The code behind the Add button: we have 3 string variables that are instantiated with the data introduced by the user. Following, we connect to the database just like we did the last time. Then, an INSERT type query is ran that introduces the 3 gathered fields into our database. At the end the text fields are emptied.

Graphical user interface, text, application, email

Description automatically generated

This piece of code is tied to the search button. This one works just like the previous button, with the exception that now we use SELECT type query, instead of the INSERT one. What this does, is that it checks all of the row’s “Nume” field if it is the same as the one that we introduced (the one that we want to look for).

Graphical user interface, text

Description automatically generated

Now, we have a checkBox. First we connect to the database, then we check if the box was check. If it is true, we run another type of query, SELECT which in this case selects all rows of the table, but this time we ask it to do that in a certain way: picks the “Nume” fields in alphabetical order. In case the box is not selected we redisplay the whole table without any sorting operations.