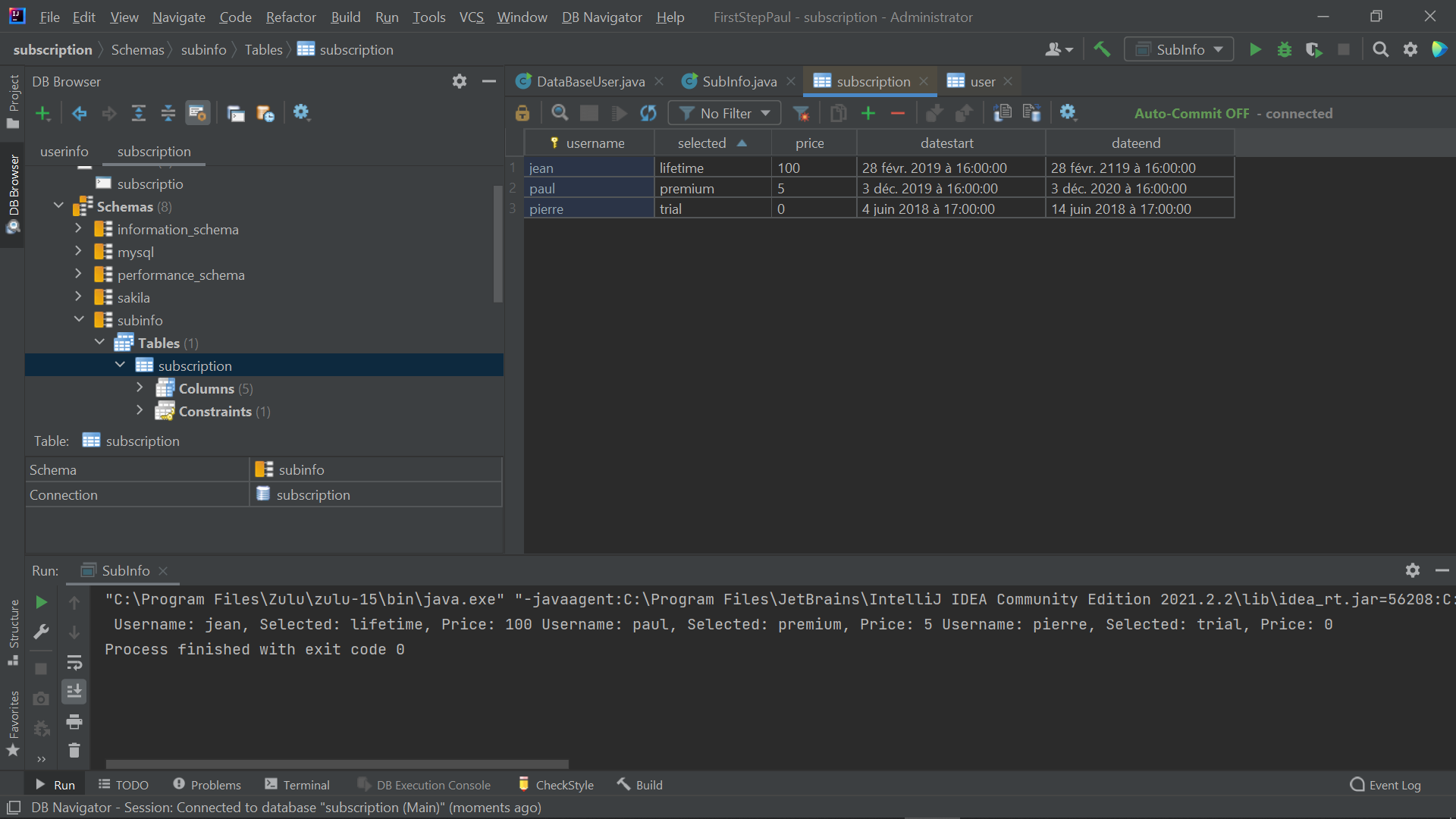
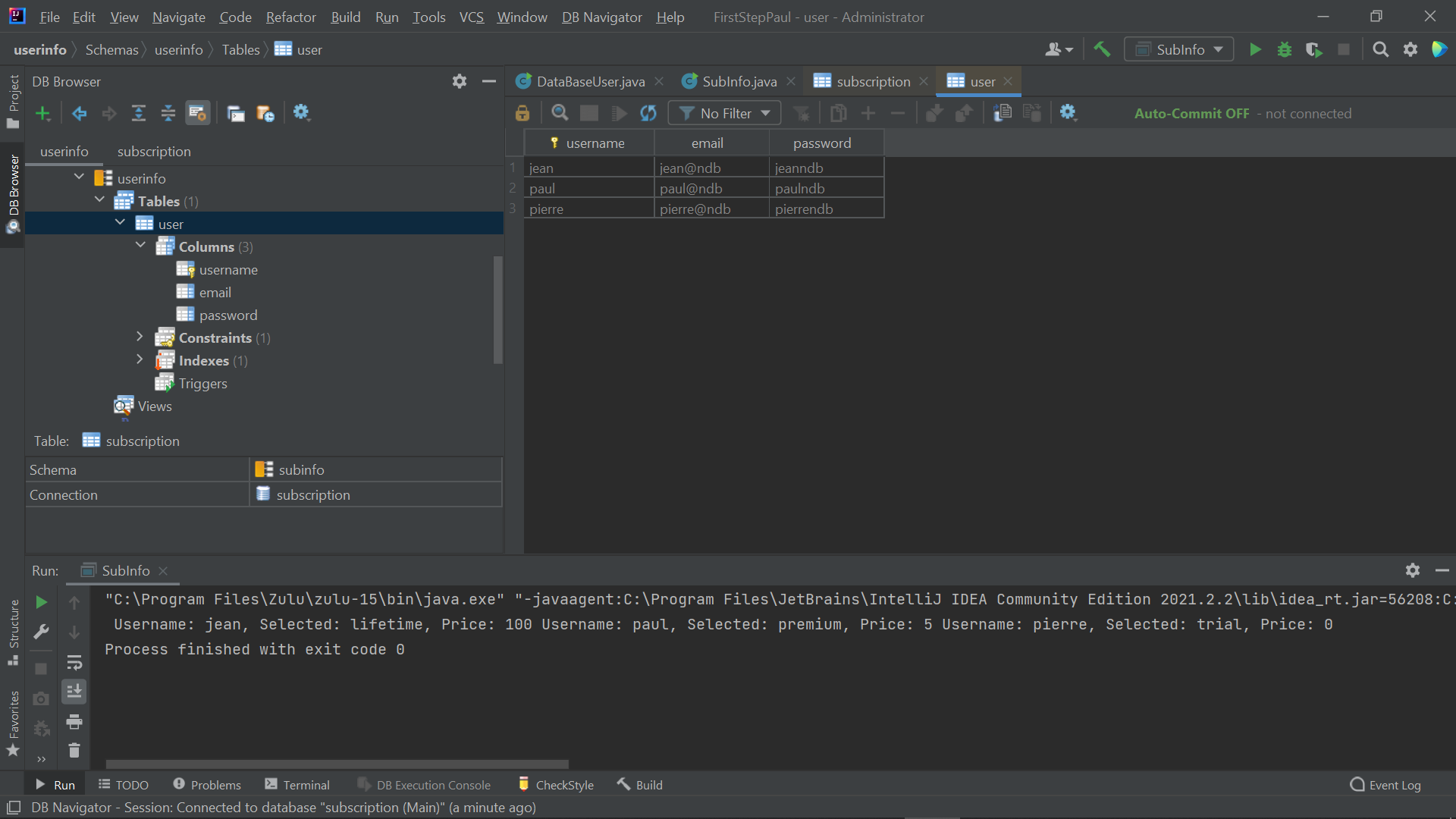
20/09/2021 :

* Installation of mysql
* Eclipse settings

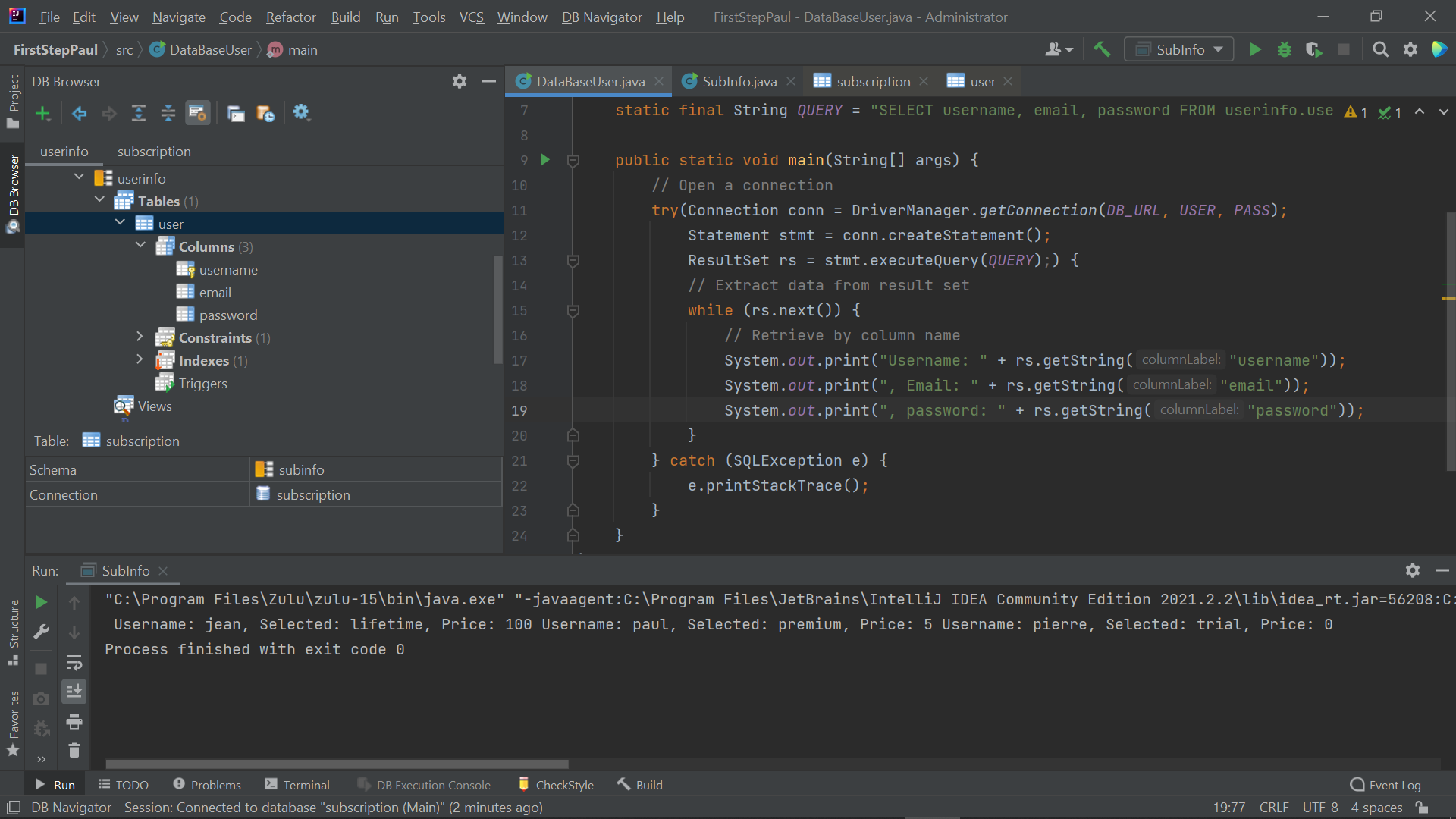
21/09/2021 :

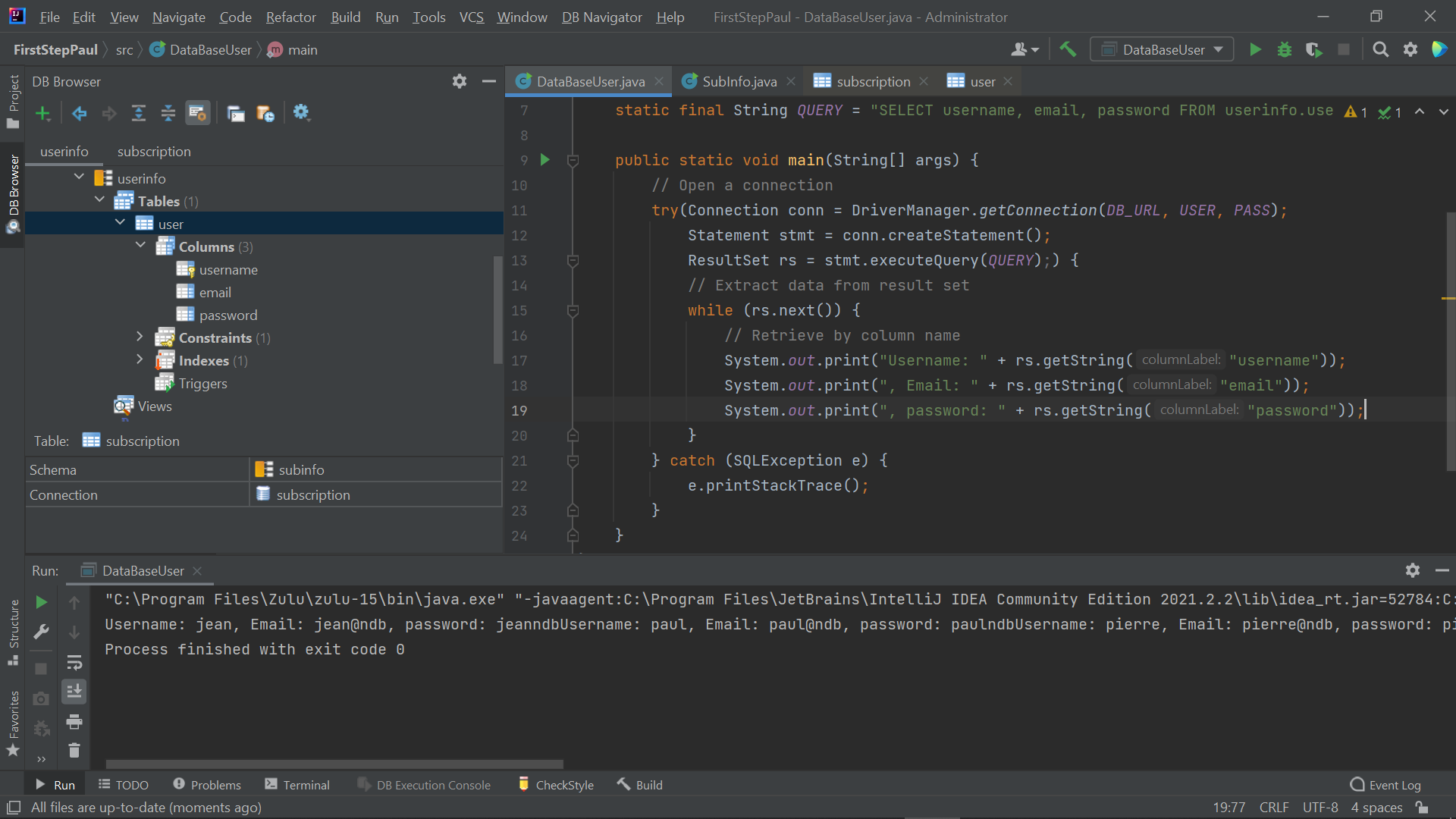
* creation of tables on mysql workbench
* installation of intelliJ IDEA
* intelliJ IDEA
* linking of databases and intelliJ IDEA
* test of the previous linking





After setting up the software, I performed 2 tests to check the connection between my databases and my classes. I was able to display the contents of my tables. The objective was to simulate user data as well as their subscription to any service.





I would like to be able to display this information in a better way so that it is more readable, for example in a column. It would be better to be able to link these two tables by the username column which would allow perhaps to do other interesting methods.

It might as well be to be able to link these two tables by the username column which would allow perhaps to do other interesting methods, like being able to choose to see the information of only one user at a time and then be able to choose to see the information of their subscription and why not change it

Code review and feedbacks:

Well done Paul, I appreciate your work. there are some notes to improve your code and boost your development skills:

1- consider a better name for your project. FirstStepPaul is not an appropriate name for a project. the name of your project should reflect the concept of your projects such as user-management or customer-services or something like them.

2- don't commit "out" or "target" or any folders that including the project outputs such as .class files.

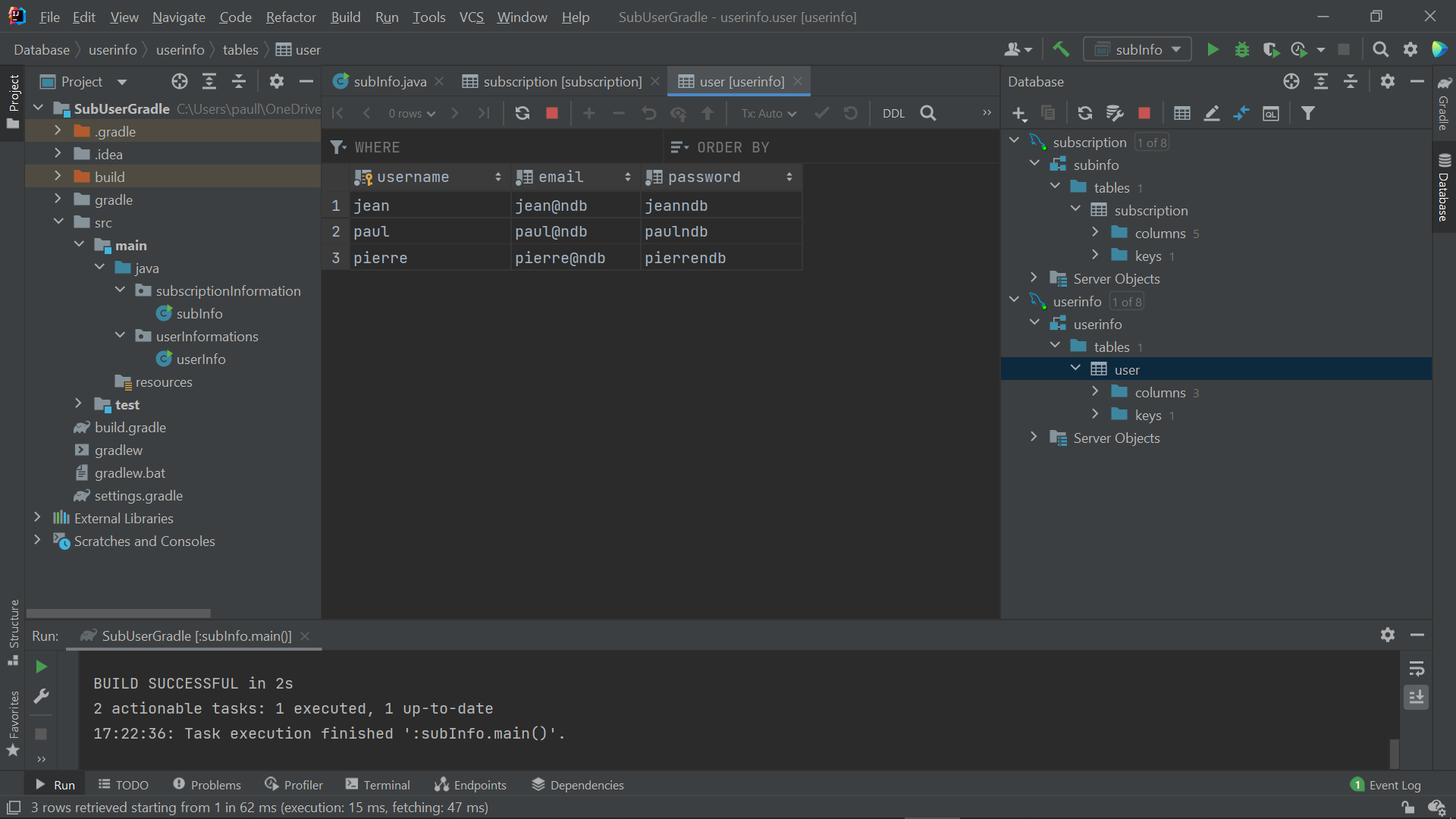
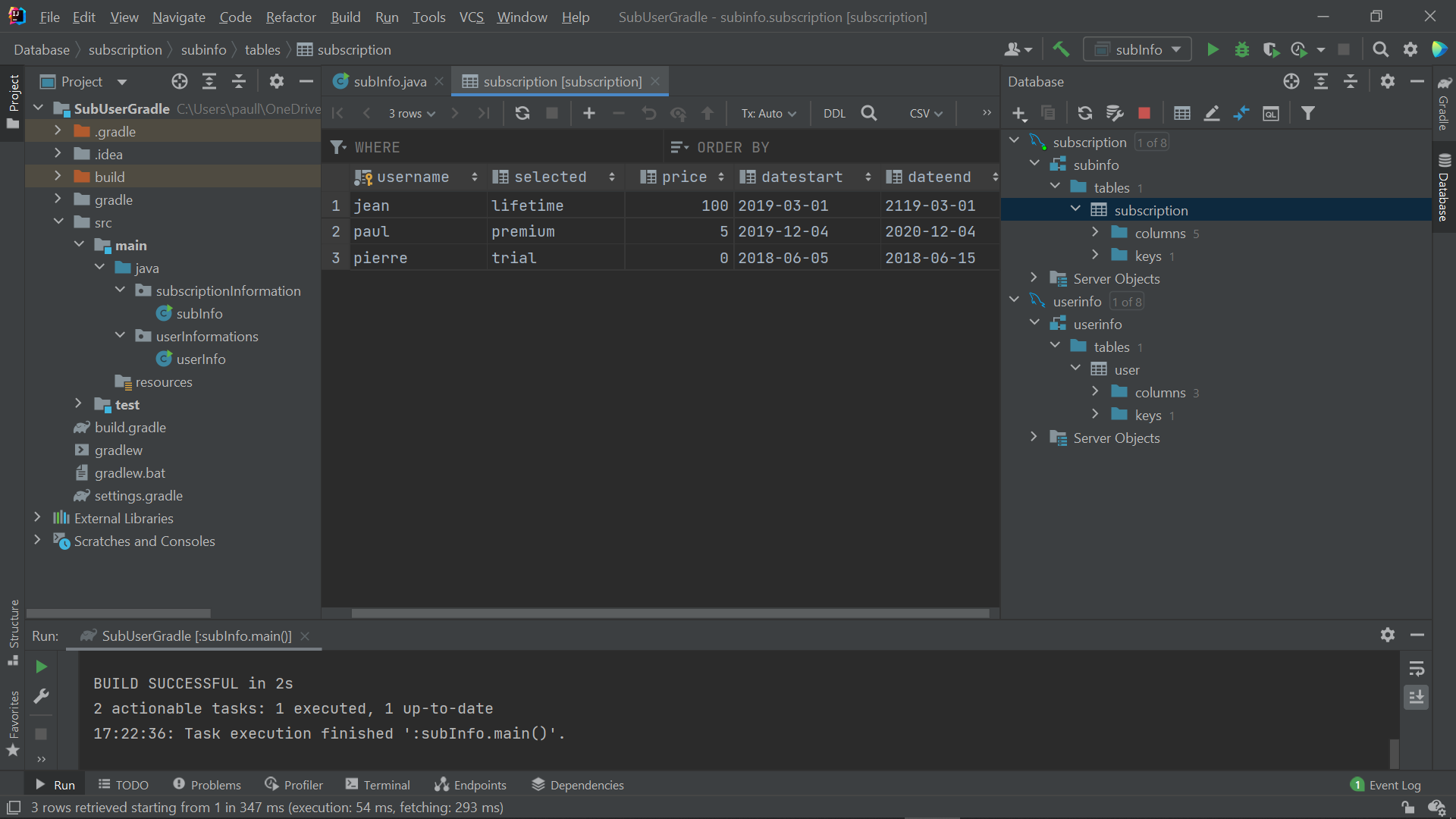
3- using the libs is not defined in your project, so if somebody downloads your project and tries to run it, face the error that you encountered. study and use "Gradle" in your project to manage dependencies.

4- try to use spring concepts (spring-core and spring-boot) as much as you can. for example for spring-boot: define the properties in application.properties or application.yml and study their usages and differences.

5- SubInfo and DataBaseUser are very similar. use Object-oriented concepts such as Encapsulation and define classes with specific duties and relevant names. then you will have a "Readable" project that everyone can understand its logic by reading it.

22/09/2021

* Installation of the ultimate version of intelliJ IDEA
* intelliJ IDEA settings
* Build a new project ( Gradle this time)
* linking of databases and intelliJ IDEA
* test of the previous linking
* Try to re-use previous code as a starting point



I had some problems when setting up my new project on Gradle.

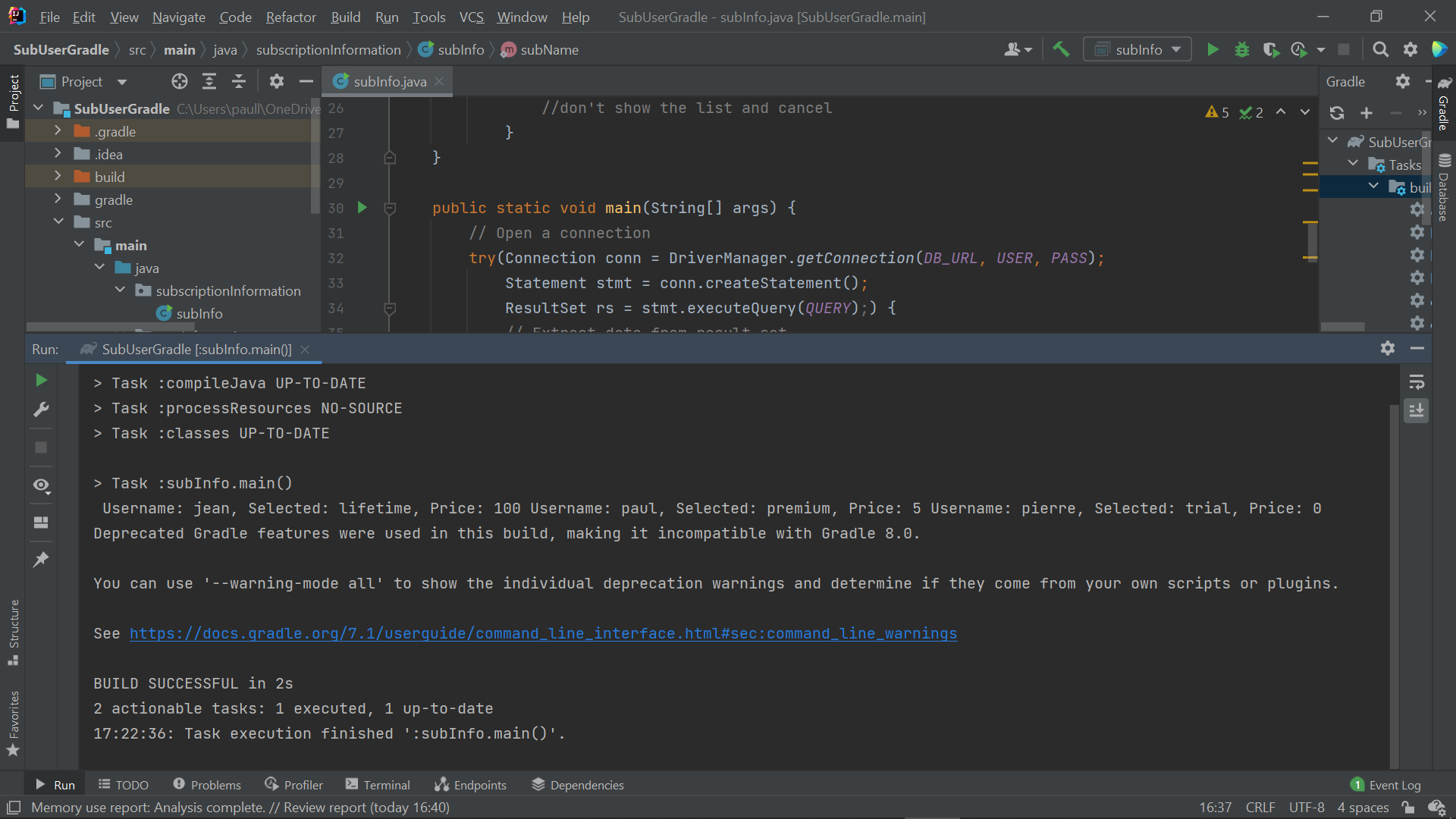
Then I followed the tutorials in the following video : https://www.jetbrains.com/idea/guide/tutorials/working-with-gradle/creating-a-gradle-project/

to see how to create it well.

I had problems with the updating of Maven repositories, I dropped the case when I was told it wasn’t important.

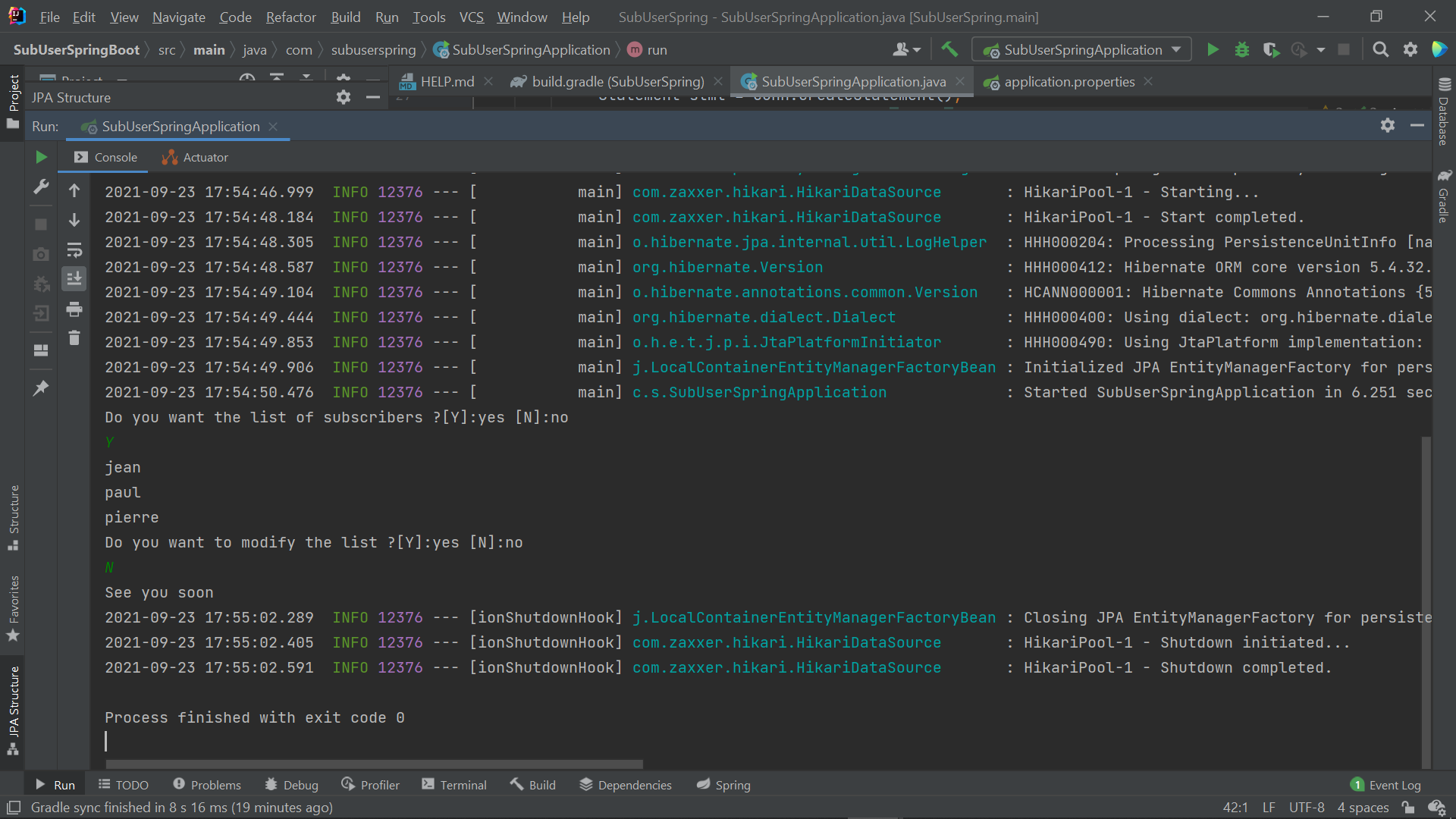
After creating packages and my old classes, when executing the code there is this error message that I can’t fix about my version of Gradle. However, the result displayed in the console is the same so I did not pay more attention than that.

I tried with the time I had left to modify one of my two classes in order to better display the rendering and especially to be able to edit the database from the console. I’m starting to see the algorithm to use globally but the hand method is a bit confusing in the implementation of the new method.



23/09/2021

* Project improvement
* creation of a springboot project
* configuration and test of the old project on springboot



After adding features to the old project, I created a new springboot project to see the difference. I configured the project directly via Intellij IDEA and searched application.properties so I could use the same project but with springboot. I added springdata but I don’t know yet how to set it up and use it.

I also researched generics types and looked at how springboot works with openclassrooms. I still have to research the jpql and springdata as well as the springboot annotations.

In the ensemble the use of springboot is better understood but I think I still have to work on it, I have not managed to modify my code so that it fully uses the spring abilities. I would look into this tomorrow as well as a little reminder about the sql keys.