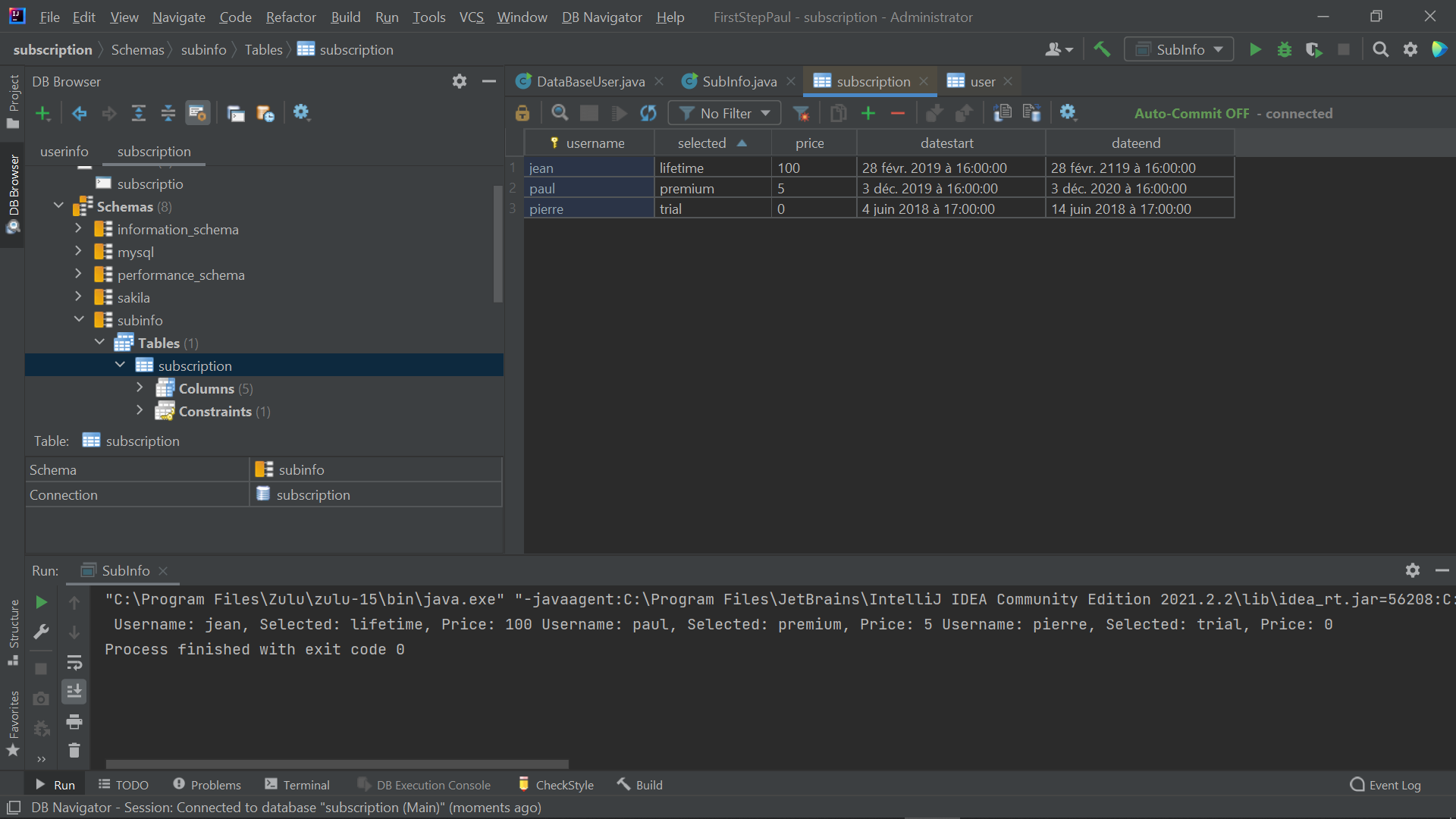
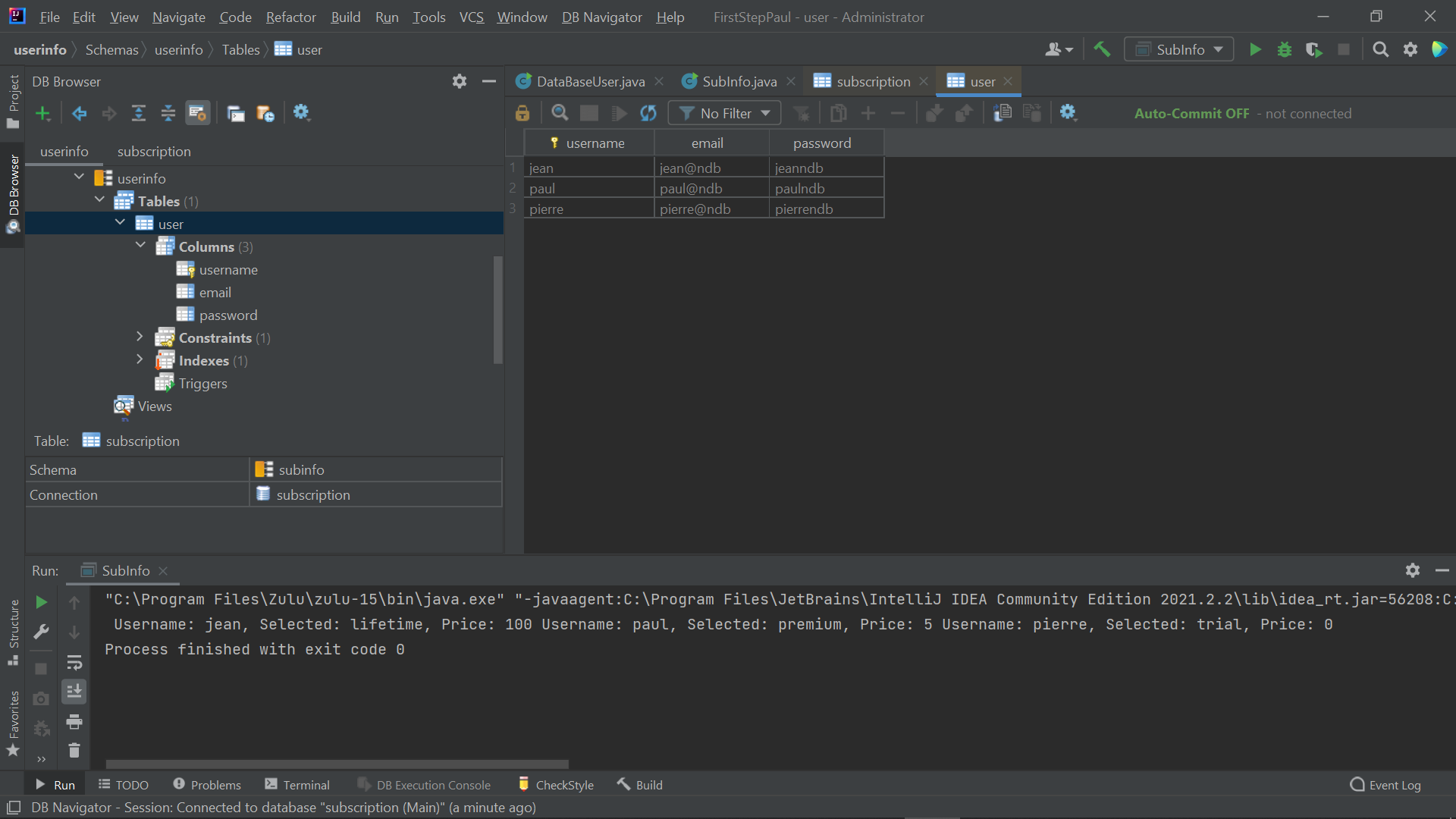
20/09/2021 :

* Installation of mysql
* Eclipse settings

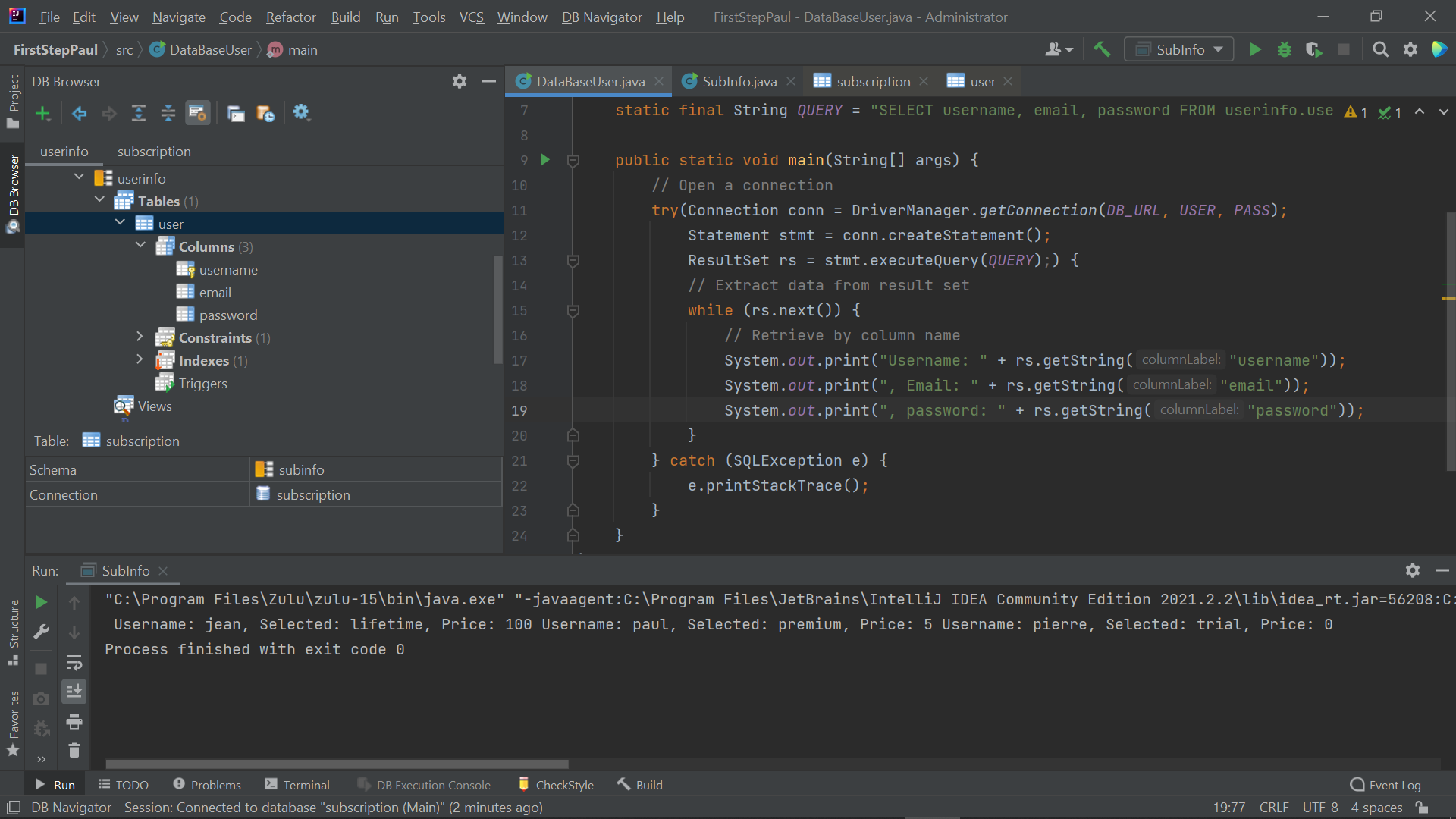
21/09/2021 :

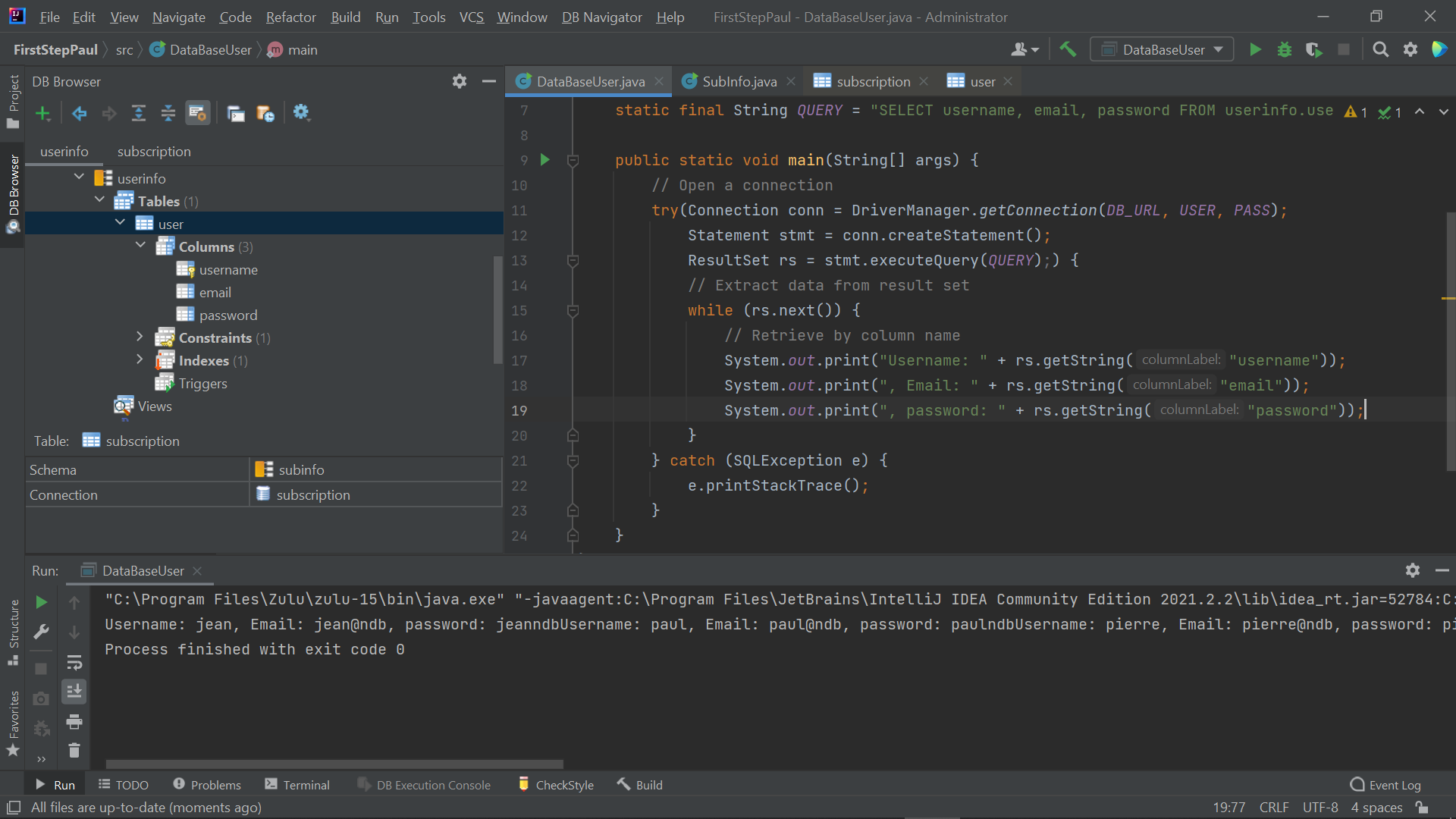
* creation of tables on mysql workbench
* installation of intelliJ IDEA
* intelliJ IDEA
* linking of databases and intelliJ IDEA
* test of the previous linking





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.

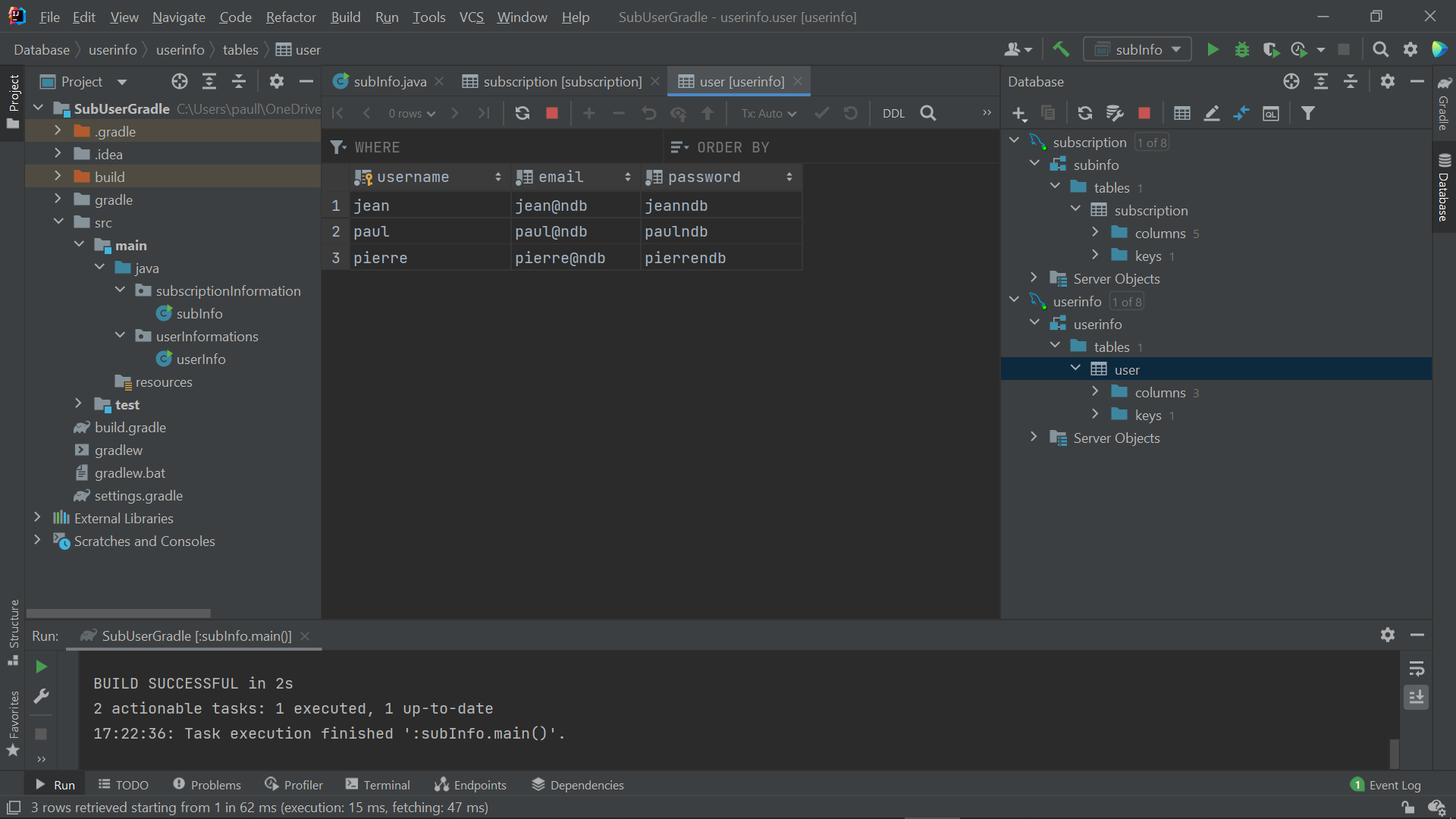
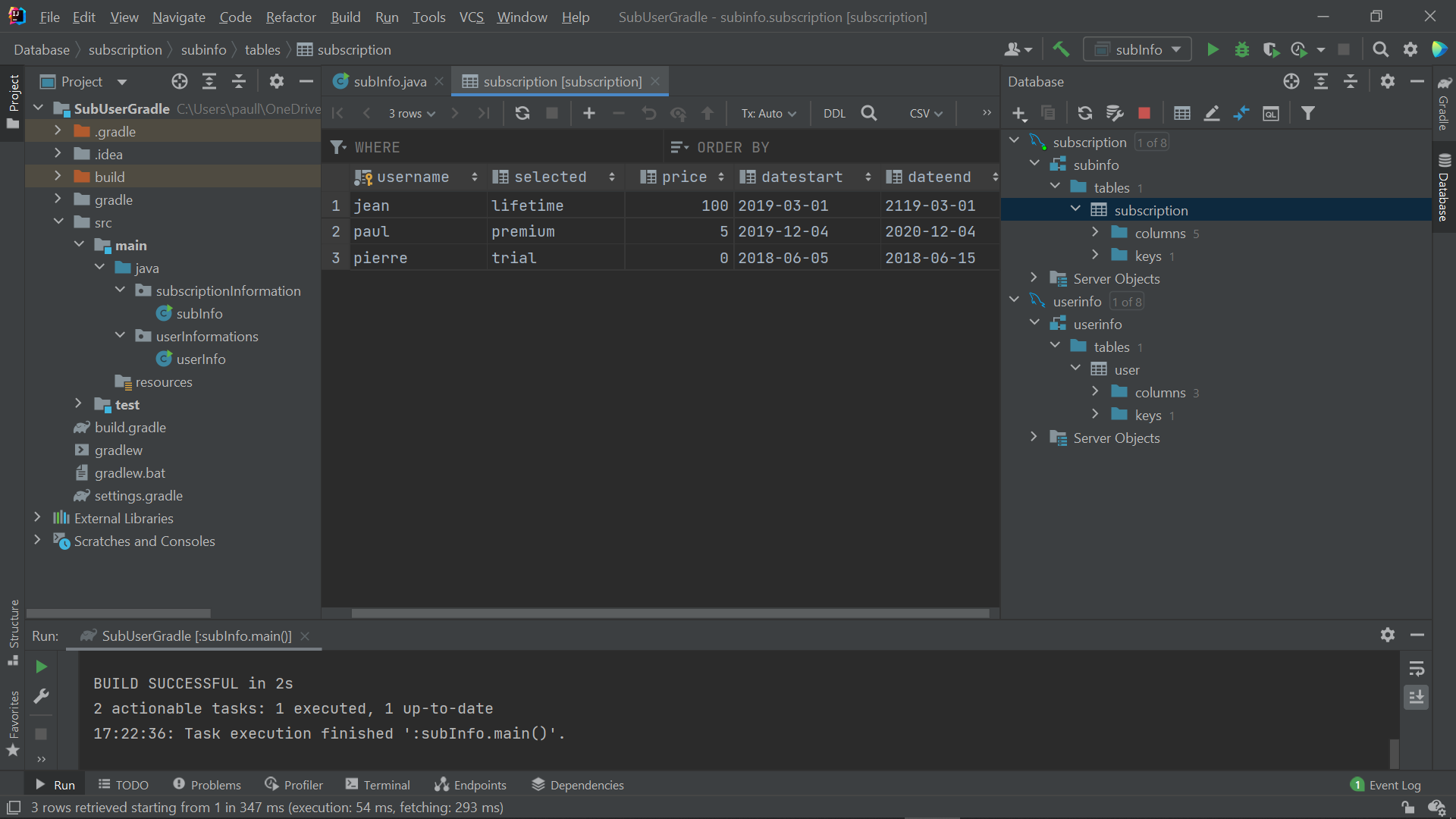
3- using the libs is not defined in your project, so if somebody downloads your project and tries to run it, face the error that you encountered. study and use "Gradle" in your project to manage dependencies.

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.

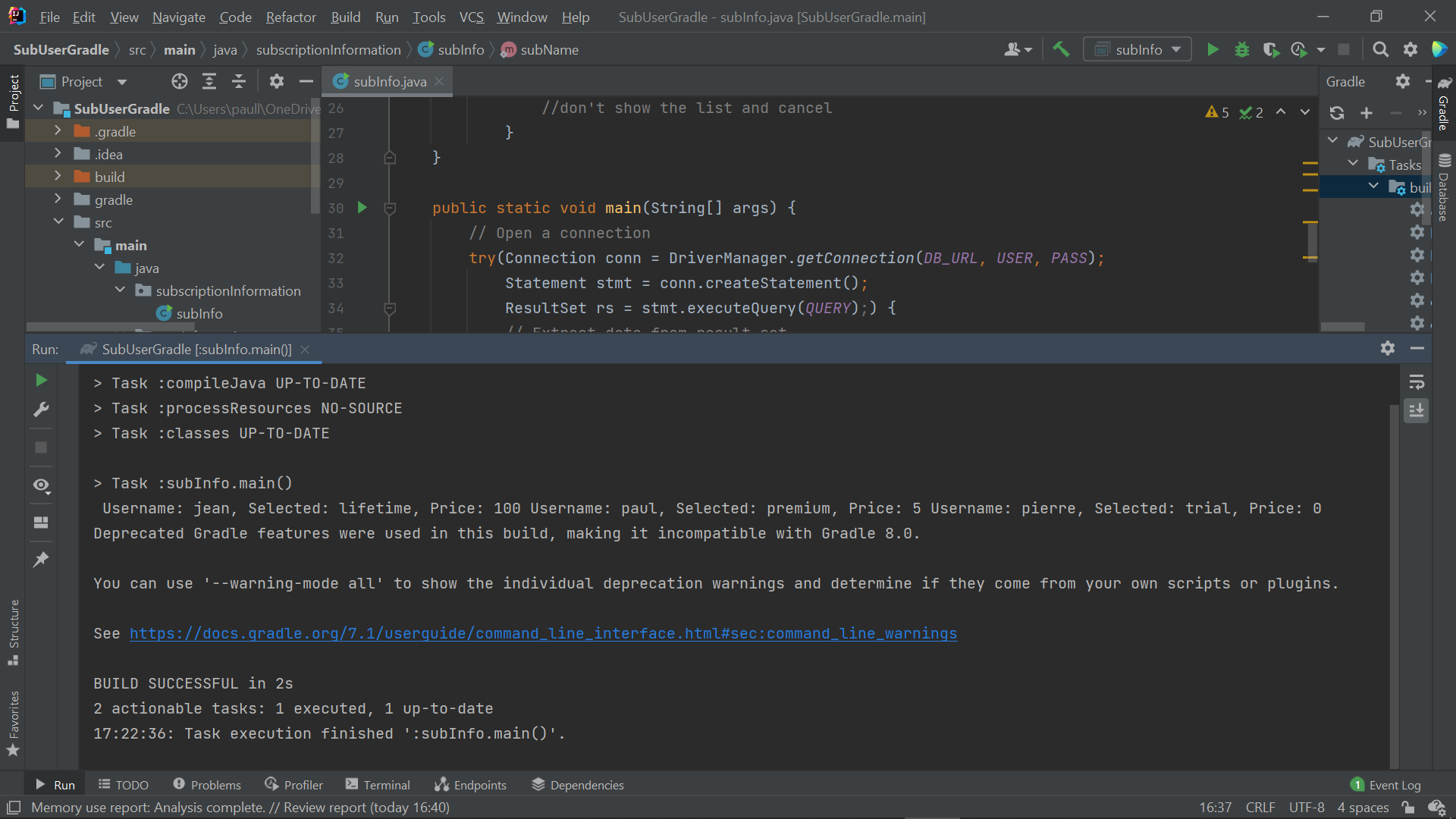
Then I followed the tutorials in the following video : https://www.jetbrains.com/idea/guide/tutorials/working-with-gradle/creating-a-gradle-project/

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.

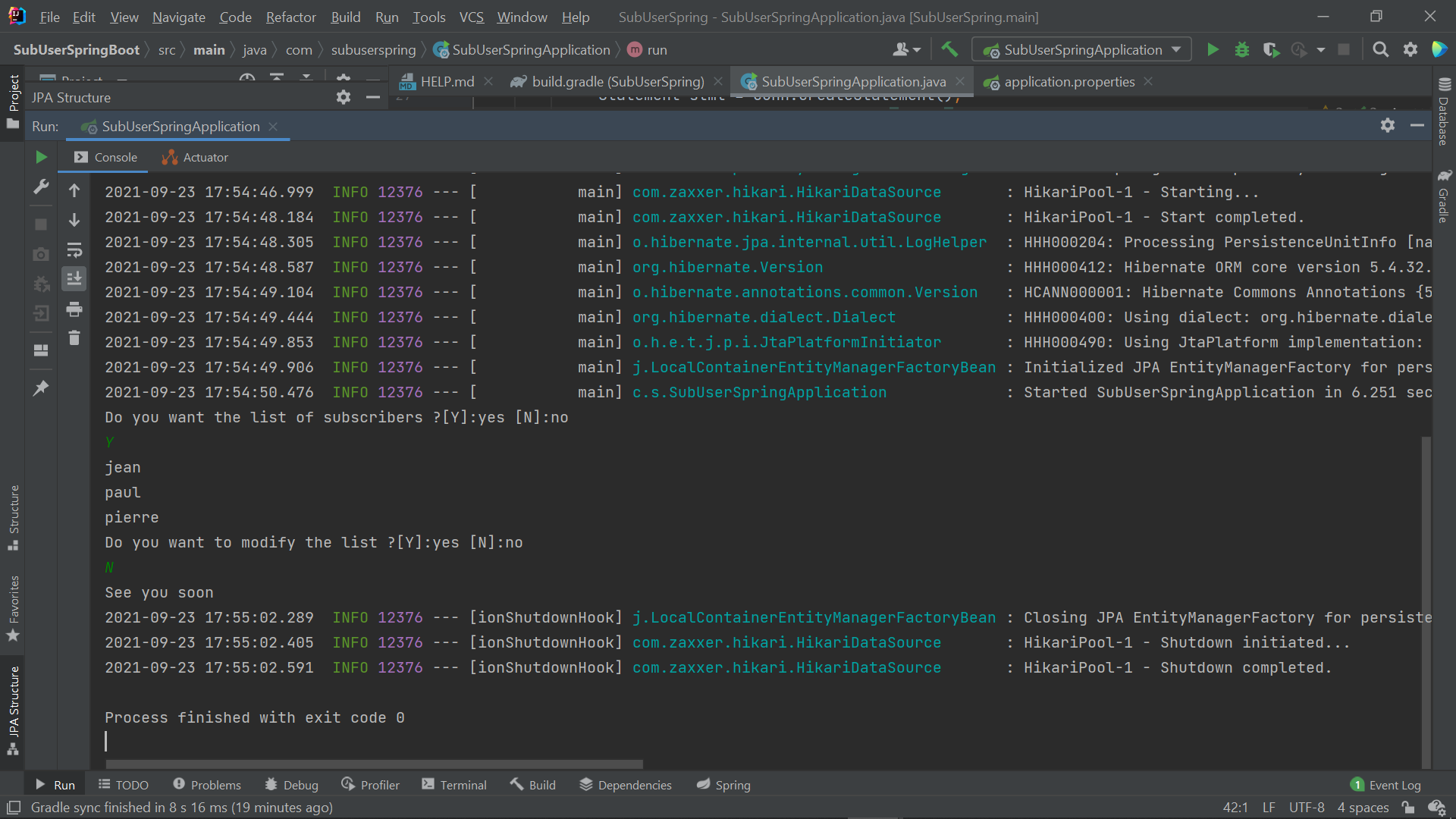
After creating packages and my old classes, when executing the code there is this error message that I can’t fix about my version of Gradle. However, the result displayed in the console is the same so I did not pay more attention than that.

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.

24/09/2021

search on the difference between jpql and mysql and on generic types. I made small projects with generic types to learn to master them, I rather adapted old projects relatively course to see what it was used for.

27/09/2021

Using JPA and small project with springdata. Lots of bean error and annotations that really slowed me down. It was better after the links sent but it is still not sufficiently mastered, still the same problems and few documents well explained on the internet I have the impression of stagnating

28/09/2021-29/09/2021-30/09/2021

I researched these documents to understand how to use the modules, as I was stuck on my project and can't resolve my concern.

<https://www.baeldung.com/spring-autowire>

<https://docs.spring.io/spring-framework/docs/5.3.9/reference/pdf/>

<https://docs.spring.io/spring-framework/docs/5.3.9/reference/pdf/core.pdf>

<https://www.petrikainulainen.net/spring-data-jpa-tutorial/>

<https://docs.oracle.com/javase/tutorial/java/generics/types.html>

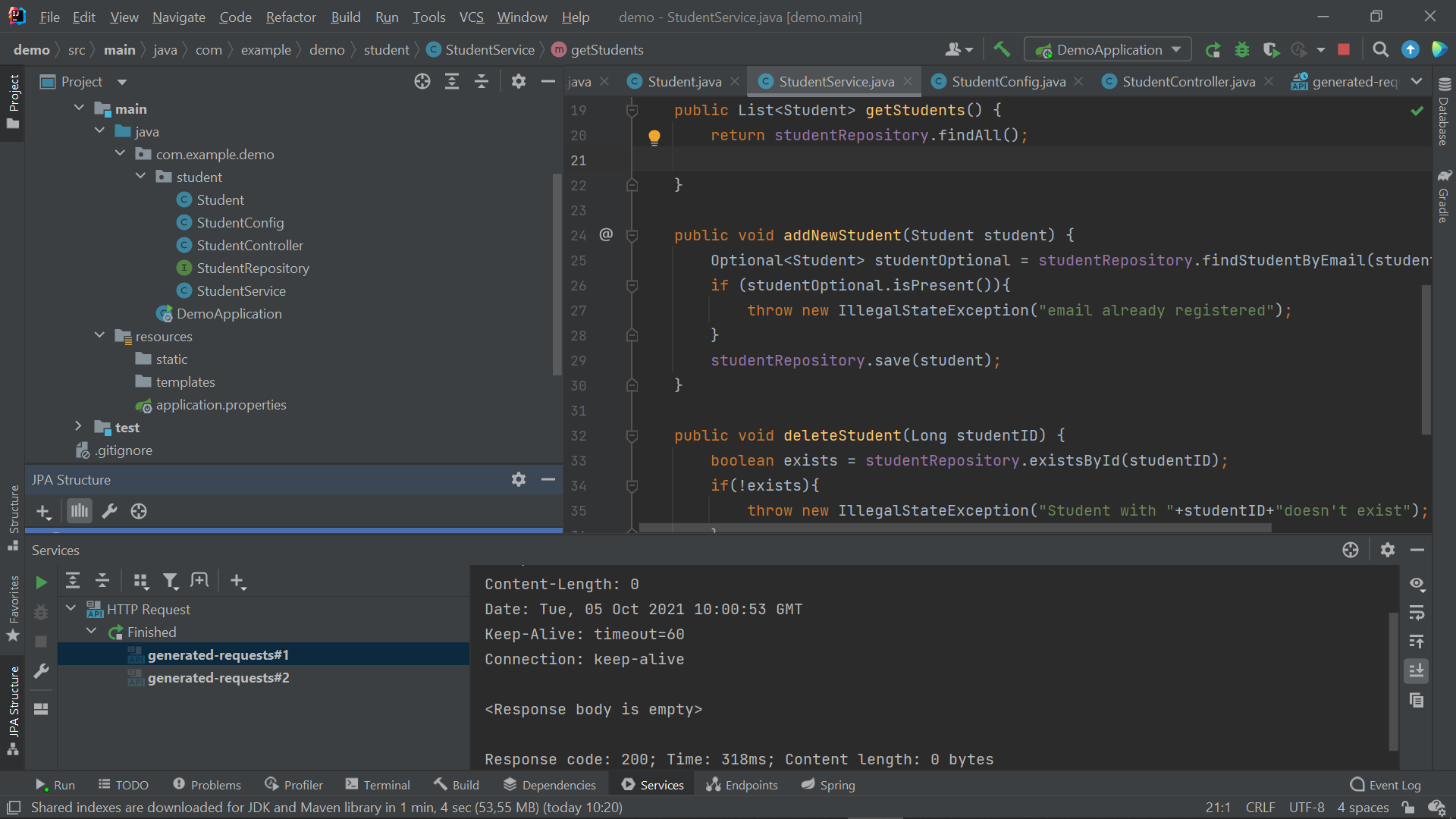
<https://www.baeldung.com/java-generics>

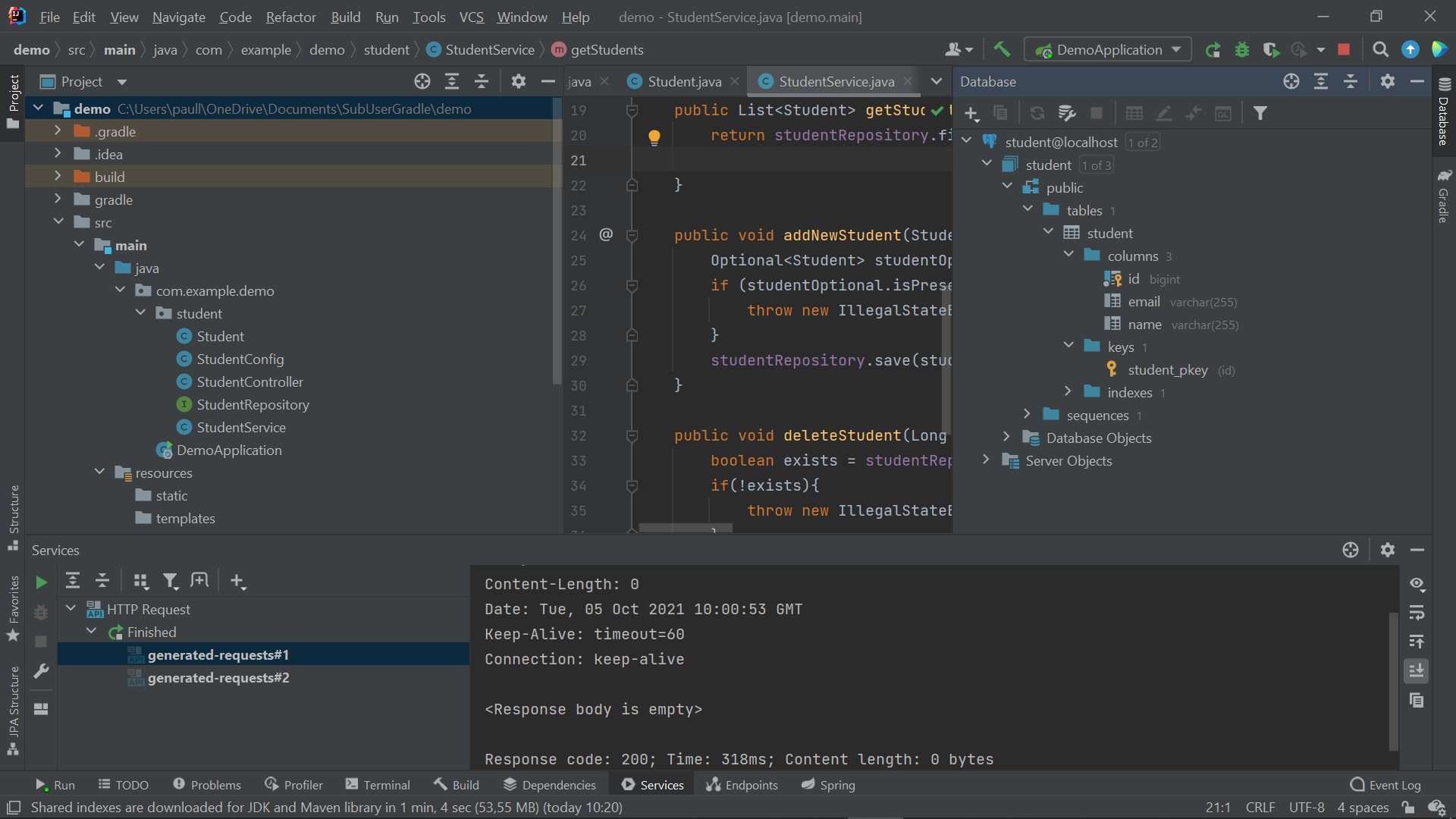
I have tried some code on intellijIDEA, but it was not really good and quiet simple, just to practice a little bit with that.

04/10/2021 – 05/10/2021

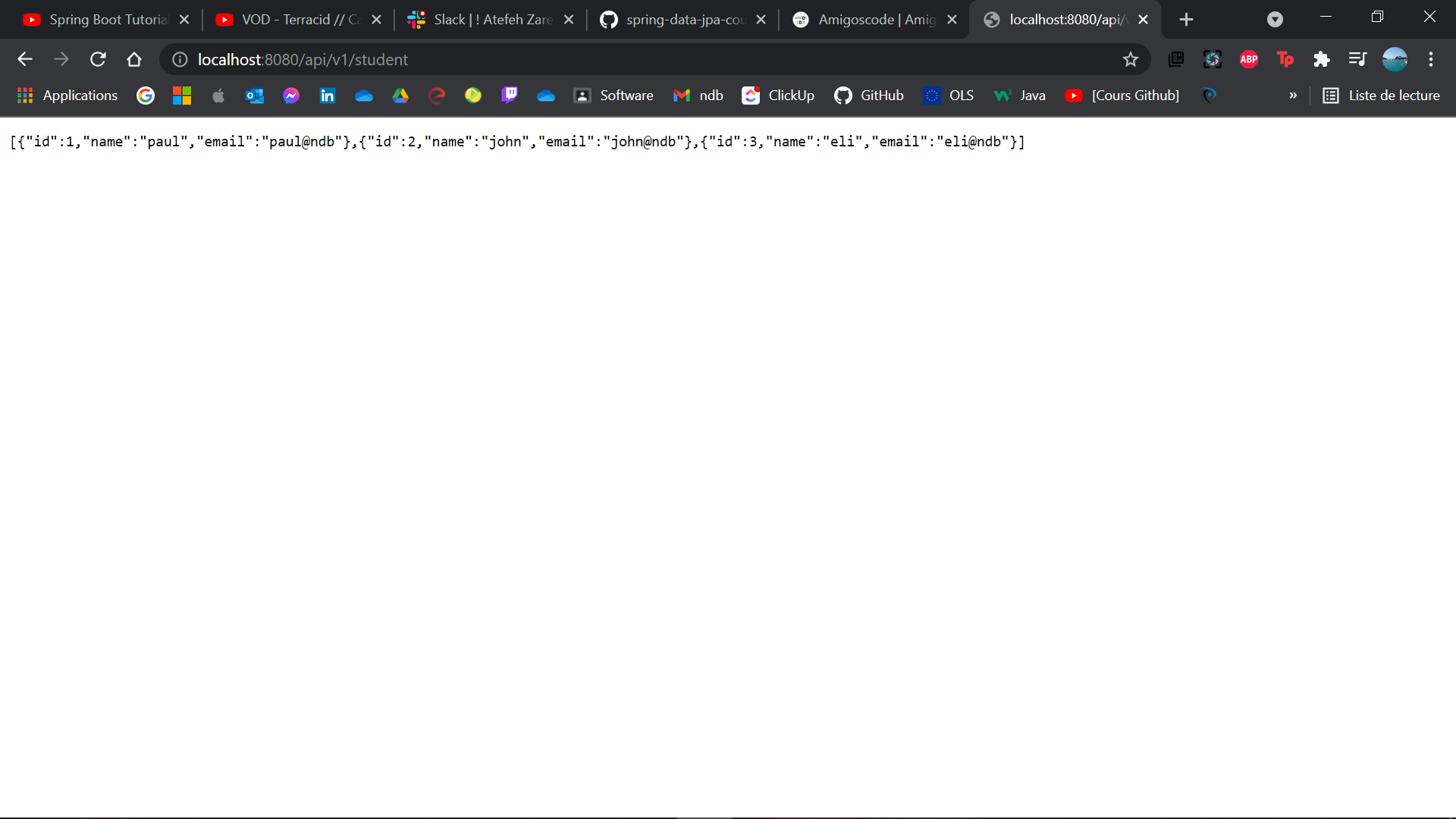
Creating a project with Spring to manipulate a database.

I have decided not to go back to my previous project and try to do a new one to learn deeper about JPA



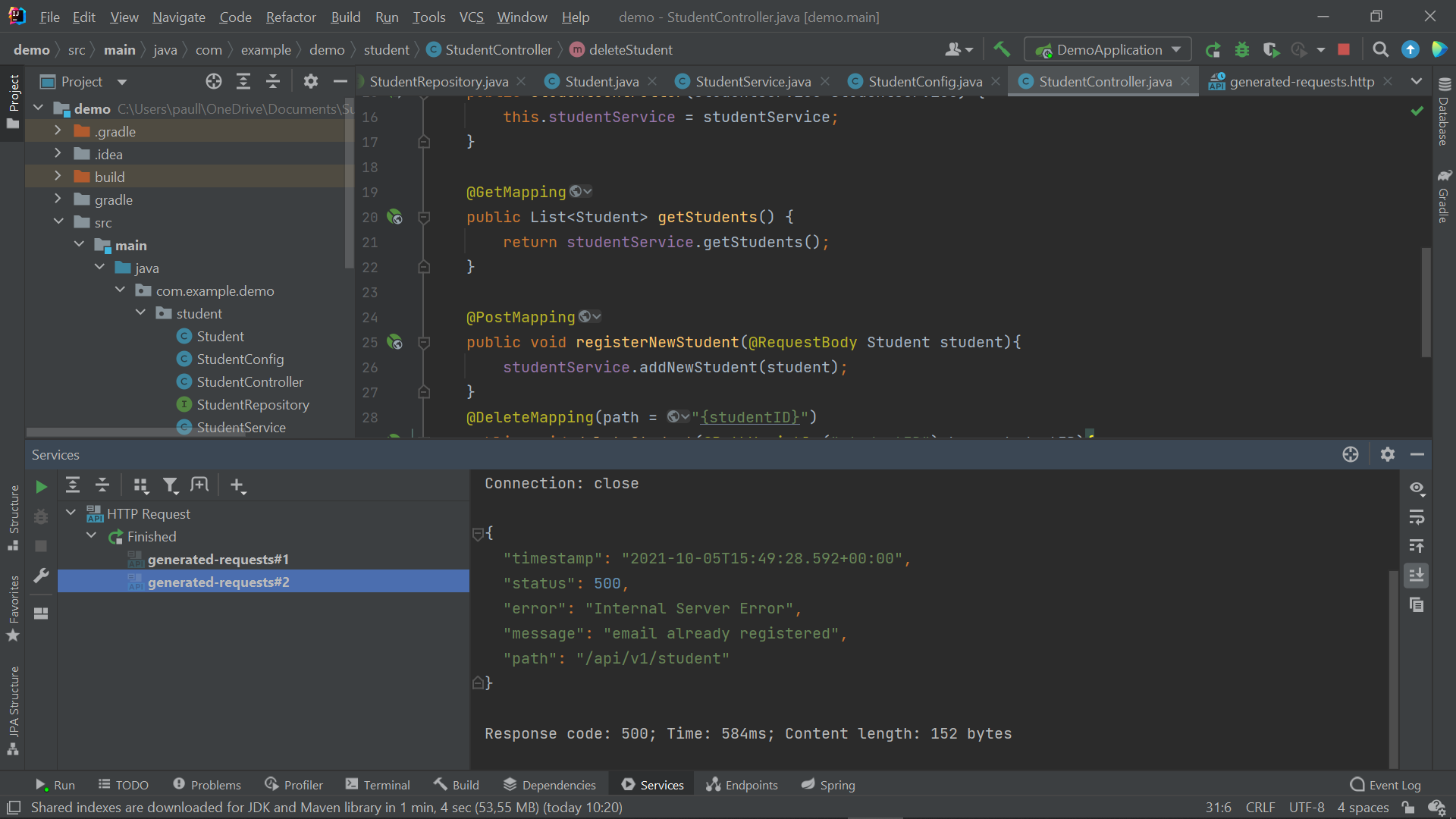


I have also learn how to use the application.properties and how to use some queries as you shown me before

I followed videos in order to create a table and to be able to display it on a local server. 

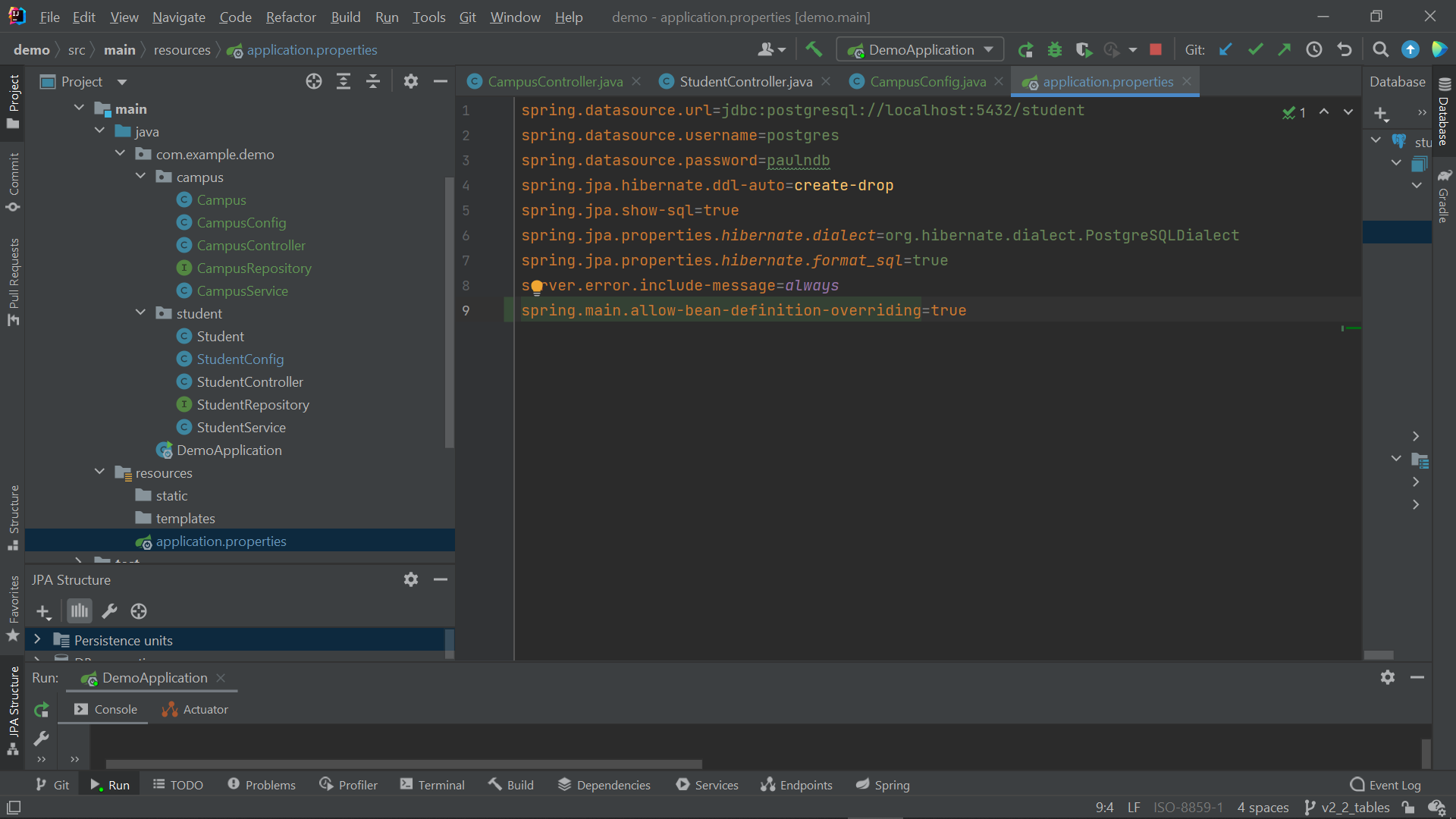
I was also able to add a person to the table if they were not already on the table and delete one if they were referenced.

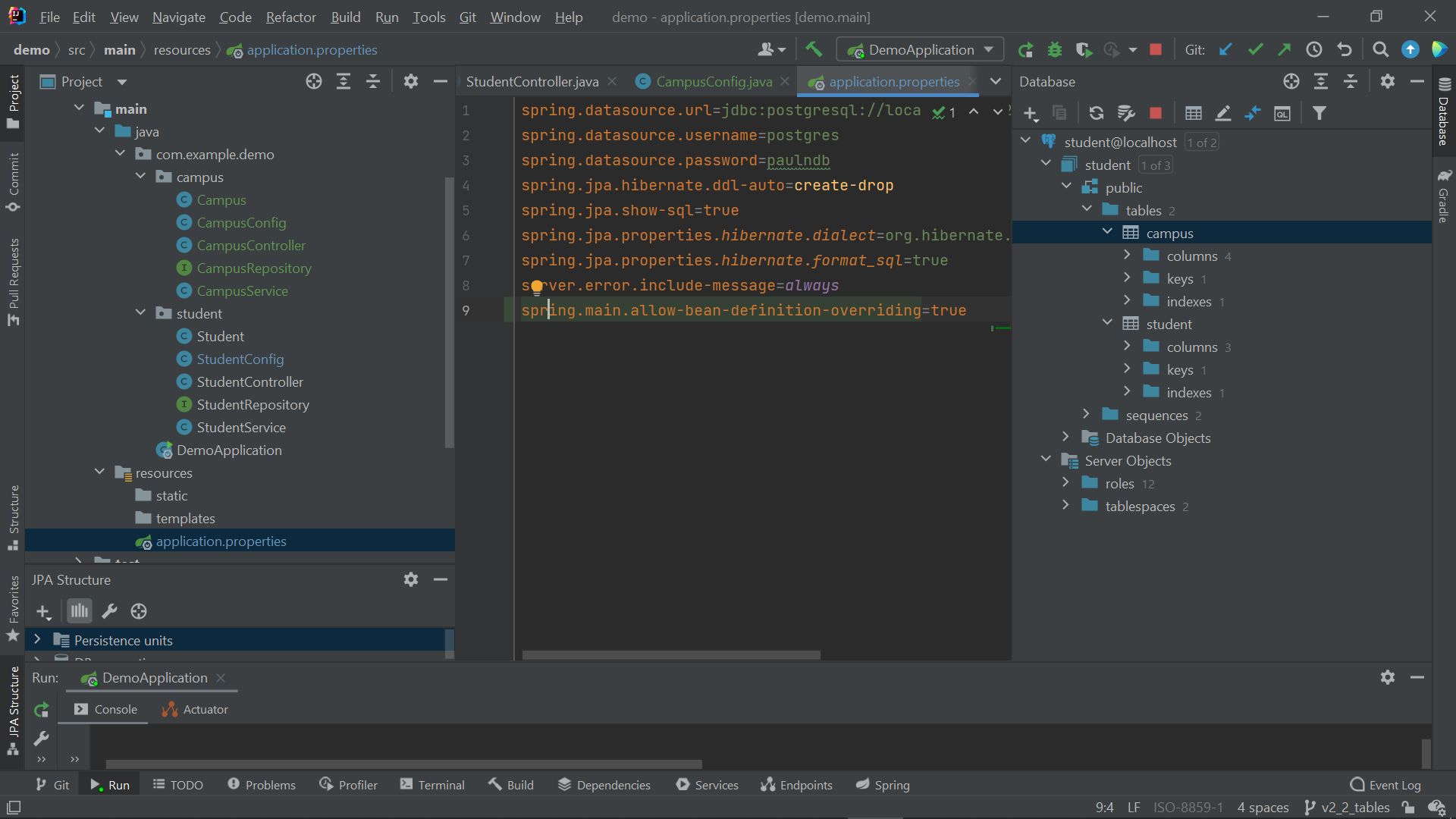


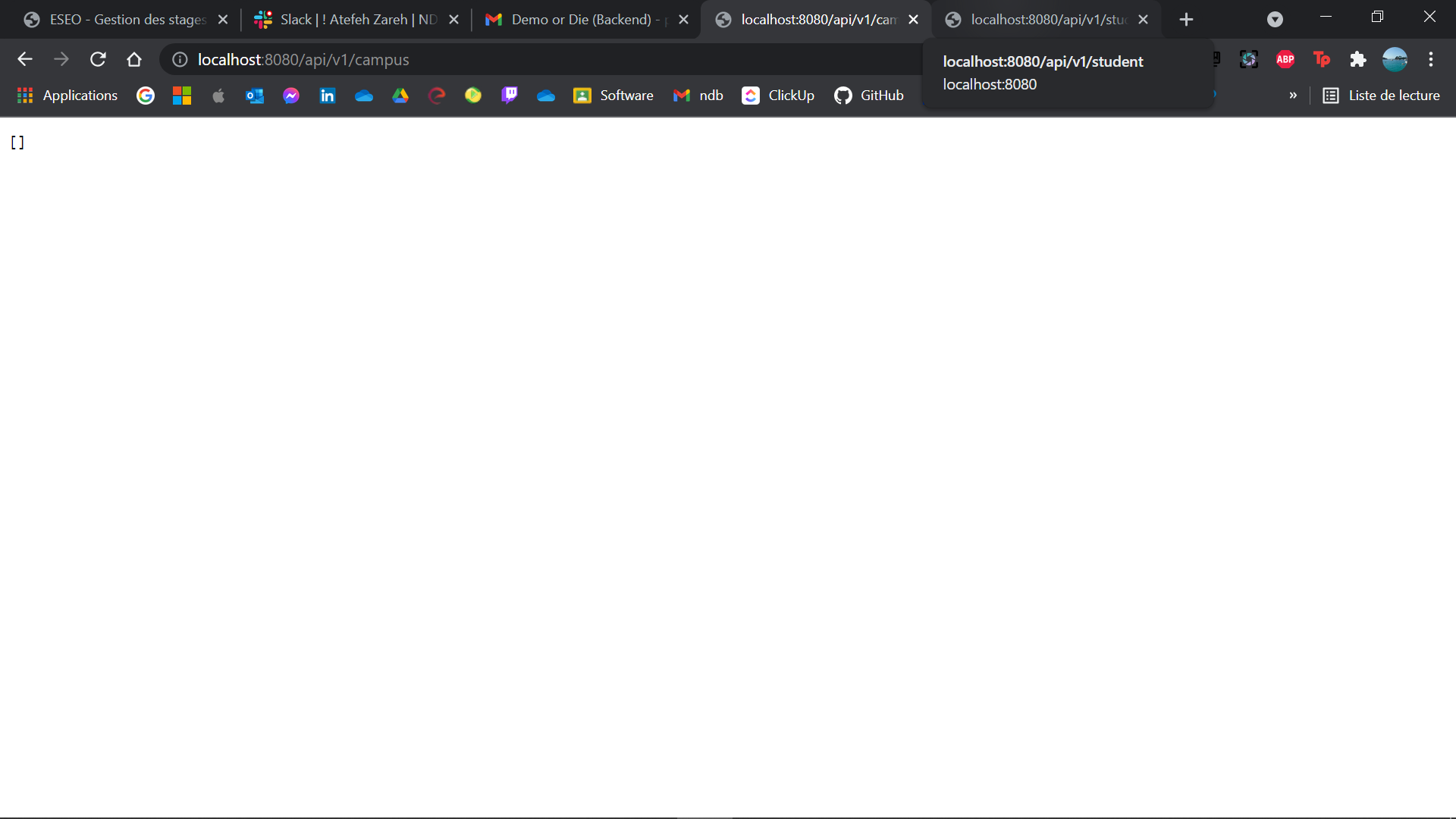


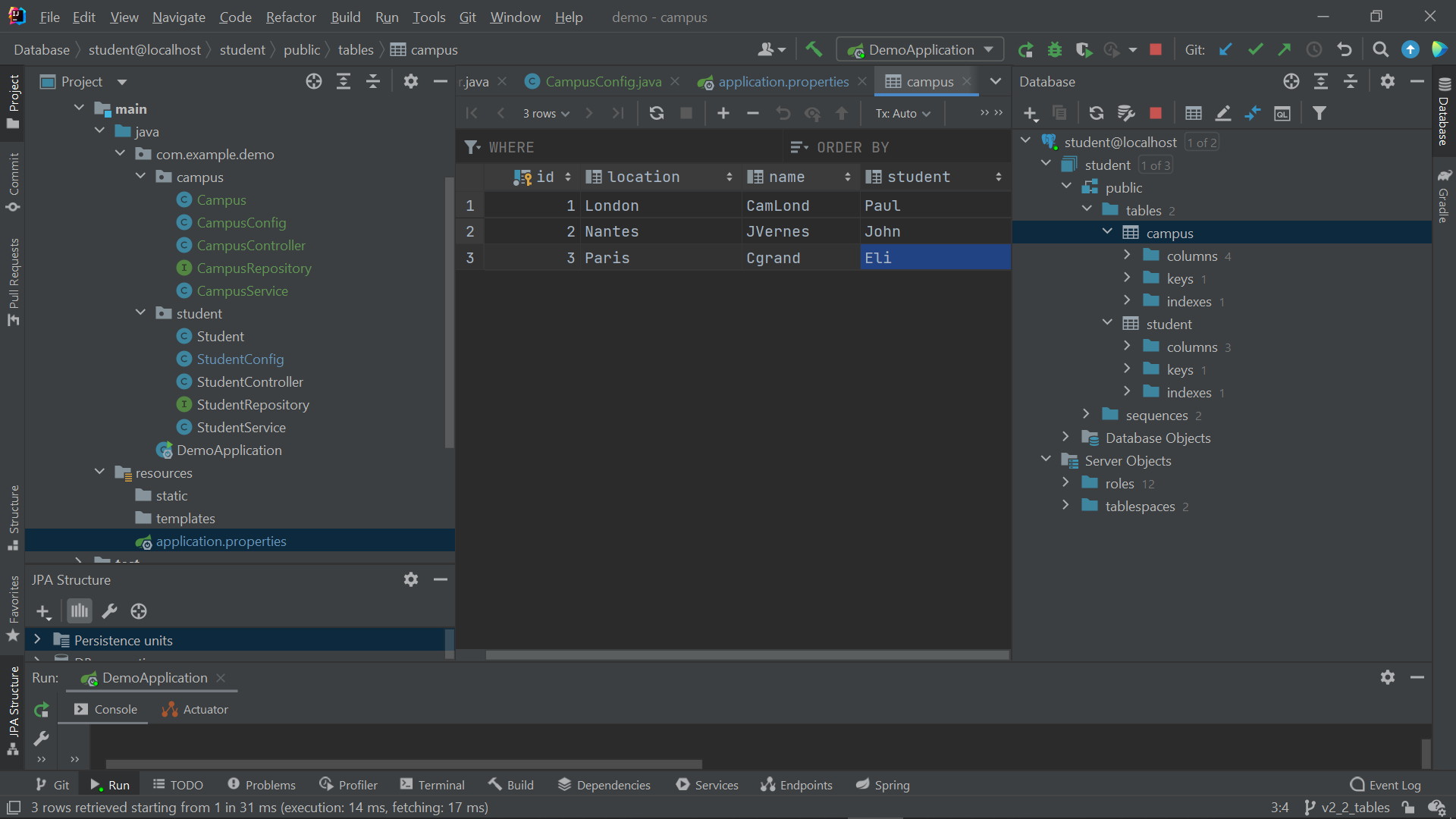
I was able to work with entities, which is rather interesting and I’m beginning to understand the usefulness of having a controller and service class to be able to work.

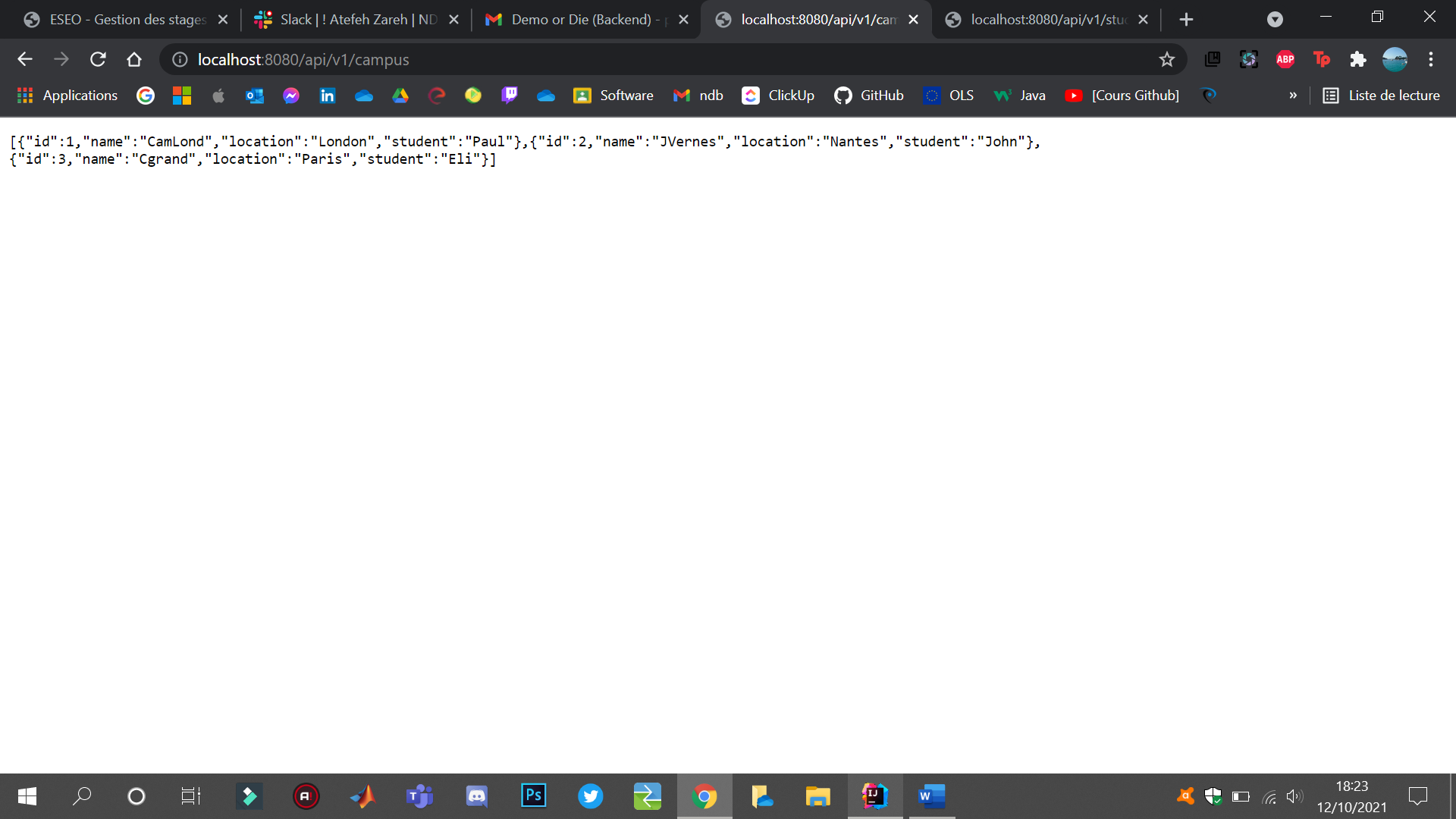
5/10/2021-12/10/2021











Creation of a second Campus table to be able to link it with the student one in order to practice the relationships between databases. Some concerns to keep information in the table

Major changes need to be made in order to have something functional such as explaining here:

*if you add data directly into your table you should use "commit" command to persist the data permanently. or if you add data with your code you should make your class "transactional" that makes the code execute in an "Transaction". Please study more about "transactional" concept and its usage in spring.*

*Additionally you should know about "Florien Key" concept. please read about it too. it is a concept in DB layer and you did not used it in your code. for example you should not mention your student name in campus, instead it is better to use its ID.*

*another thing is that you used the relationship in reverse format. instead of referring the student in the campus table you should refer to campus in student table. it means you should have campus id field in student class. because in your code every students can join more than one campus but any campus just can have one student. it's not correct in real word. right? think about it.*

*in the wright format that I said, it is reverse and every student can join just one campus and every campus can have many students. it is a one to many relationship example. assume you have campus 1 and campus 2 and for student (studentId, name, campusId) = (1, Paul, 1), (2, Nima, 1), (3, Giorgi, 1), (4, Mary, 2), (5, Alice, 2):   it means the you and Nima and Giorgi are in campus one and Mary and Alice are in campus 2. right? let me know if you understand oneToMany concept. and Please stuudy this concept by just searching about "one to many relation in DB" and "many to many..." , "one to one...".*

*please add another class for reading a file and add these data in file format too. for example this format would be ok and you can have some minor changes too:*

*add this data to file and process it in your code:*

*Campus:  
id, name, location  
1, Greenwich, London  
2, Vancovor, CanadaStudents:  
id, Name, family, tel, email, campusId  
1, Paul, Ladmirault, +017867867, Paul@ndb.technology, 1  
2, Nima, Golsharifi,* [*+987987987*](tel:+987987987)*, Nima@ndb.technology, 2  
3, Giorgi, ?, +67576576, Giorgi@ndb.technology, 1AcademicProfile  
id, studentId, level, major  
1, 1, A, Computer Science  
2, 2, B, Computer Engineering  
3, 3, C, Marketing  
Courses:  
id, Name, semester  
1, Programming, Summer-2020  
2, DB, fall-2021Attendance:  
attendanceId, studentId, courseId  
1, 1, 1  
2, 1, 2  
3, 2, 1  
4, 3, 2  
5, 3, 1*

*the relationship between (student and campus is manyToOne), (student and academic profile is OneToOne) and (student and course is manyToMany).*

*for implementing the manyToMany you need a "join table", in this case Attendance is our join table. please think about them and study about them and let me know if you have any questions.*

*and make sure that you update you daily report EVERY DAY and just commit your code everyday and put the description of your new practice too.*

*it would be great to help you to learn how to work with these very essential concepts and would be helpful for you always.*

13/10/2021

One to many relationships:

<https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP_Help/one-to-many-relationships.html>

Many to many relationships:

<https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP_Help%2Fmany-to-many-relationships.html%23>

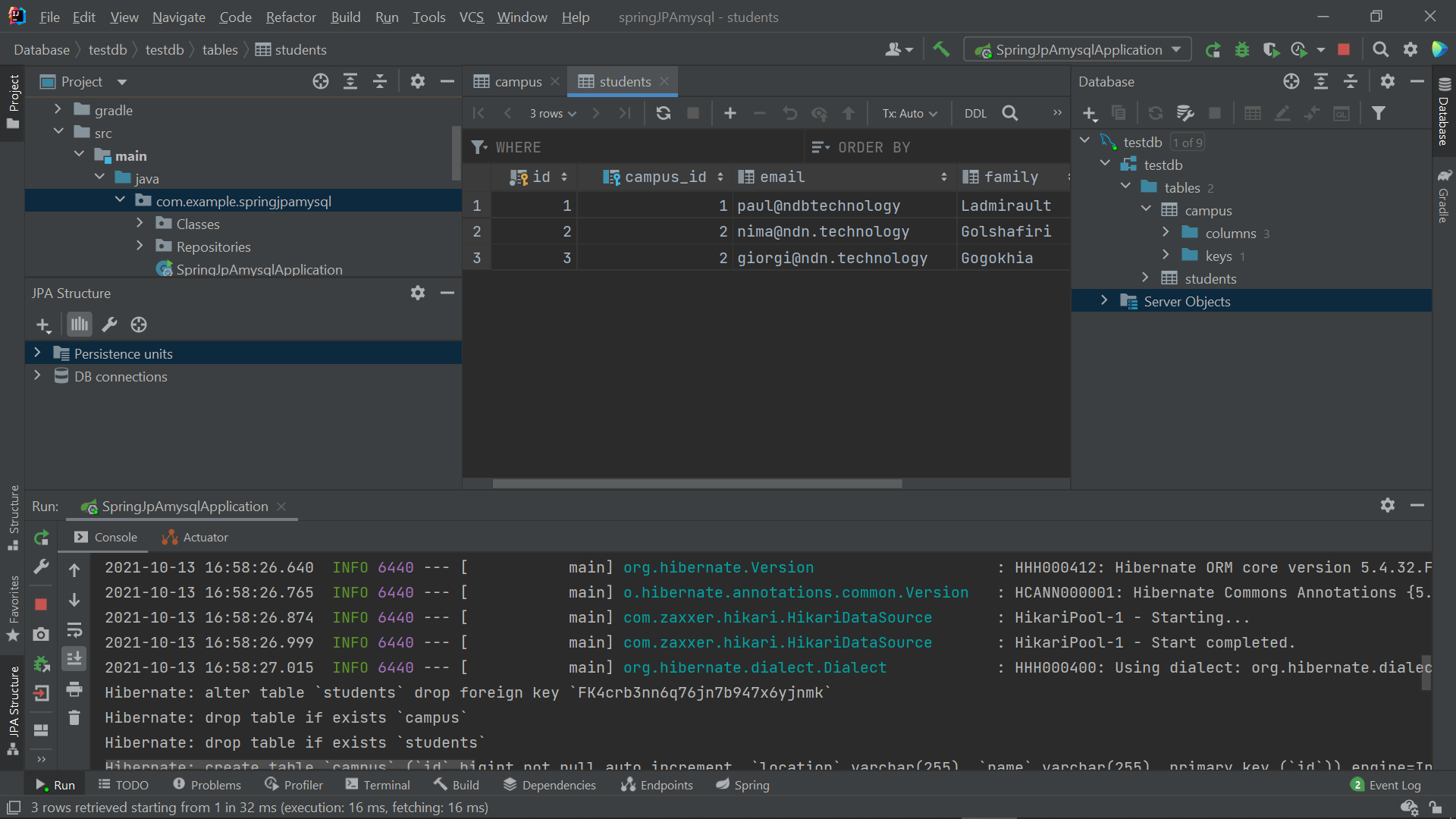
First video about one to one but with h2 db so I will not use it:

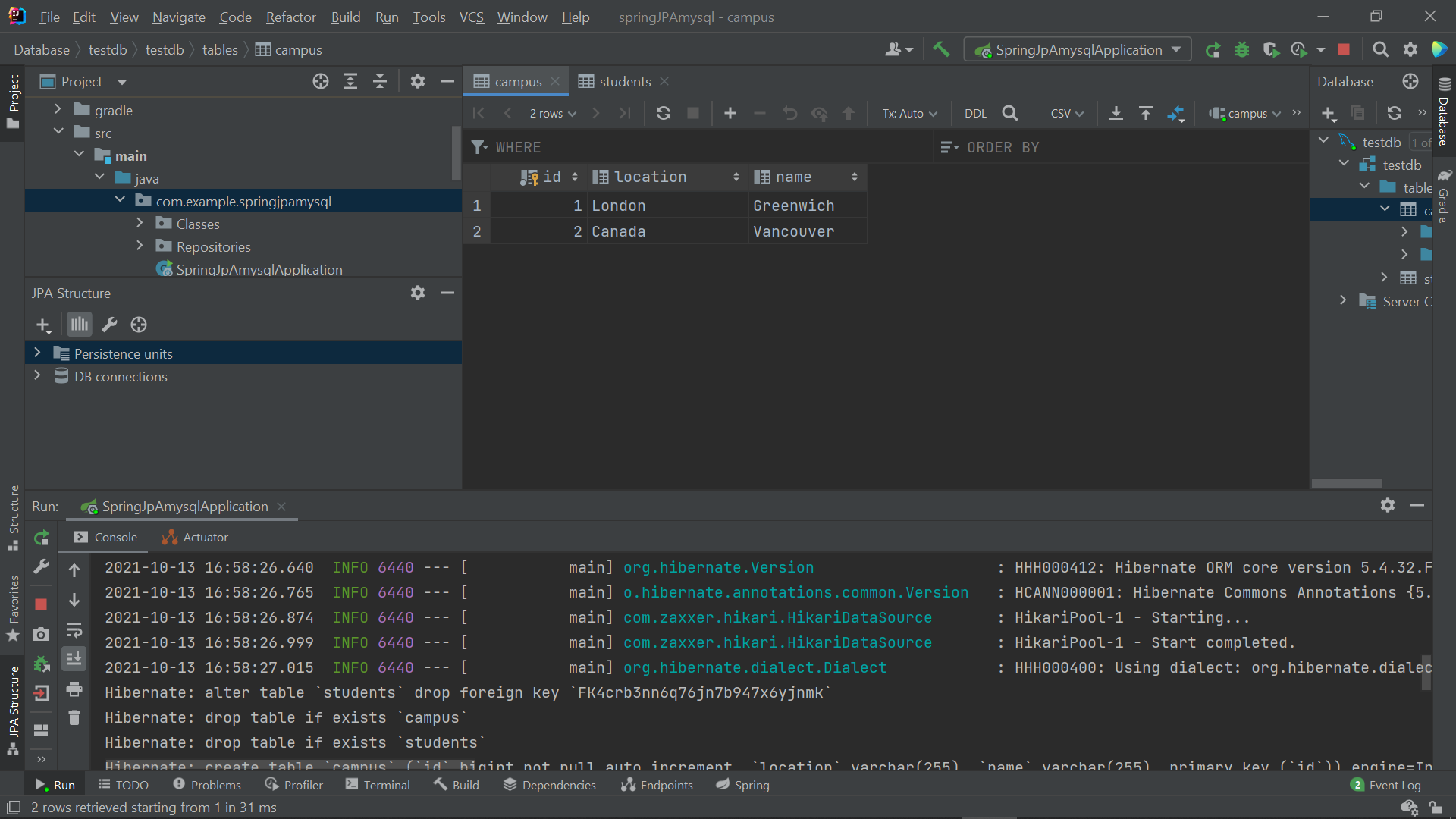
<https://www.youtube.com/watch?v=tSb02fMEB5o>

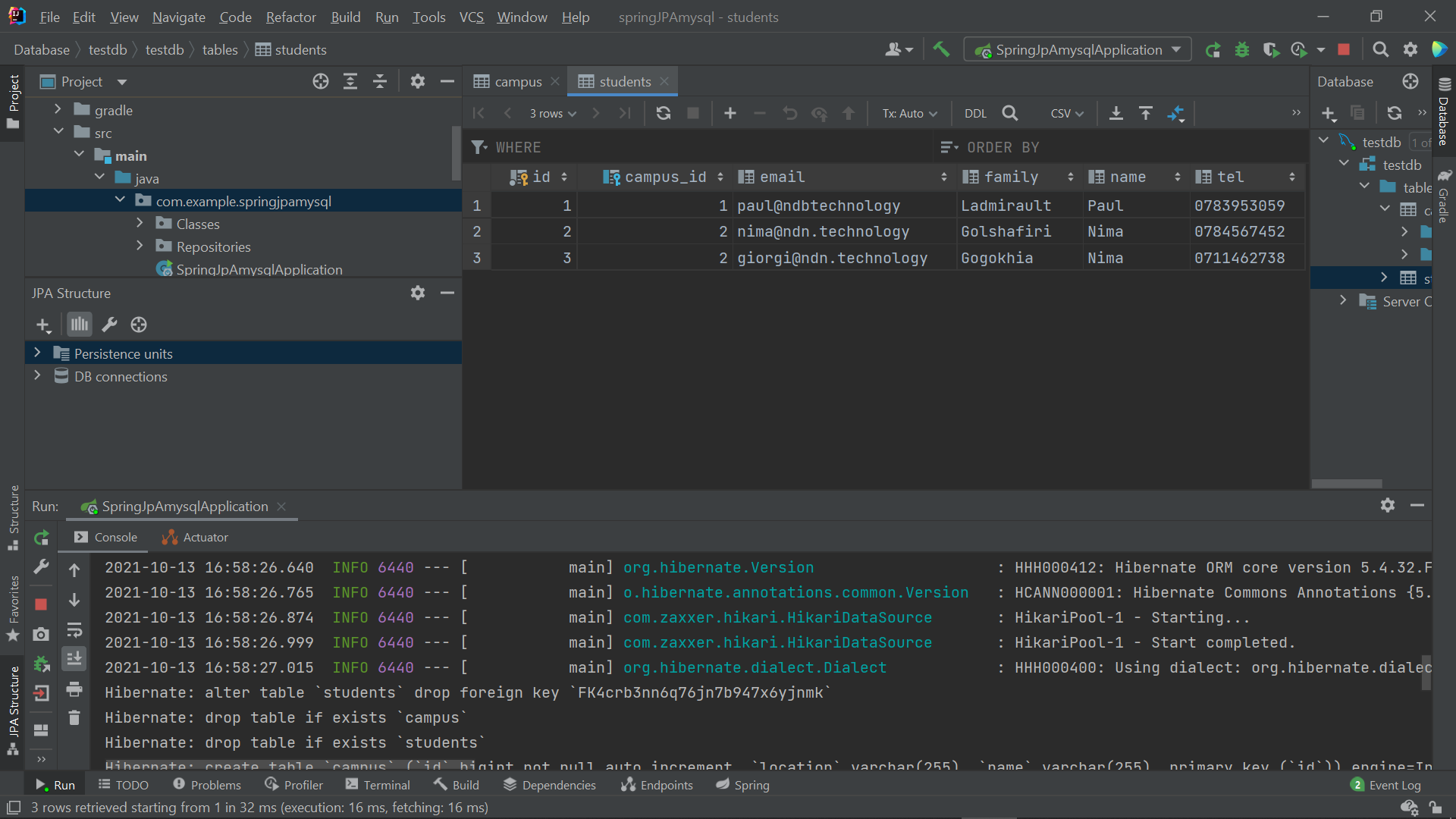
With mysql :

<https://attacomsian.com/blog/spring-data-jpa-one-to-many-mapping>

project:







Here we have 2 tables link by the foreign key campus\_id it is the OneToMany relationship that is what we wanted to do today.