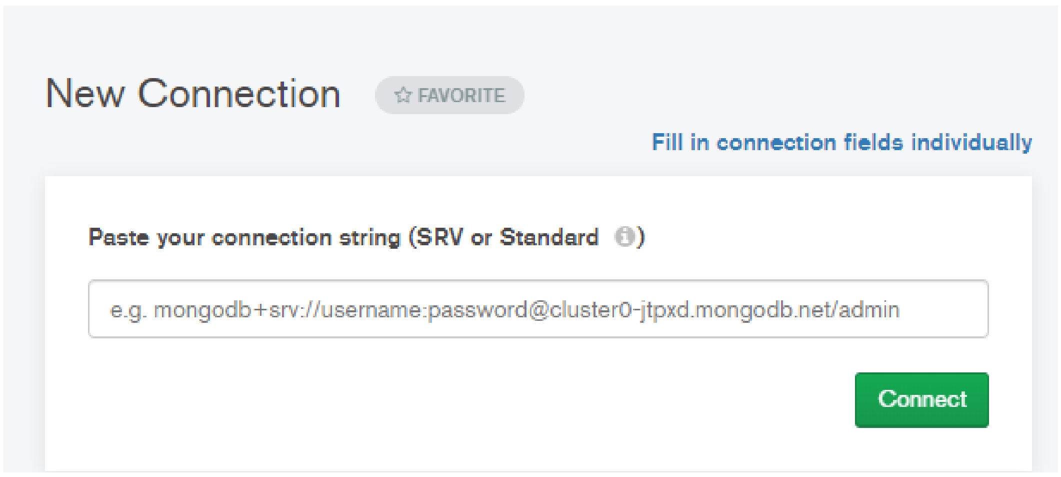
Lab 8

Part A:

Copy and paste the Bank application from Lab 5 part C.
Modify the new Bank application as such that all methods of the AccountService class are transactional. You can do this by placing the @Transactional annotation on the class instead of the individual methods.
Now modify the domain class Account and make all relationships LAZY.
If you run the application everything should still work.
Then remove the @Transactional annotation and notice that you get errors.
Then add the @Transactional annotation again.
Write in a document why it is that we don't need eager loading anymore and the Application still works with lazy loading.

Part B:

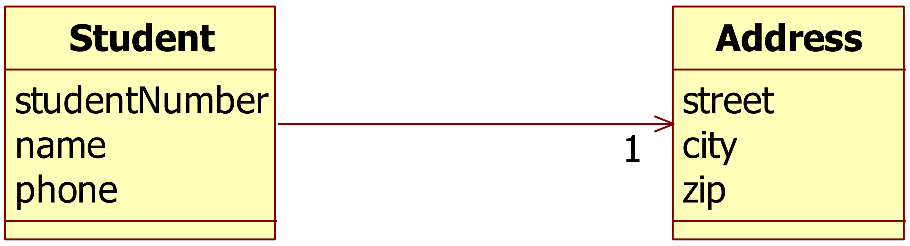
First, we must start the Mongo database by running Mongo Compass (shortcut access is on the Desktop "**MongoDBCompass**").


New Connection
FAVORITE
Fill in connection fields individually
Paste your connection string (SRV or Standard)
e.g. mongodb+srv://username:password@cluster0-jtpxd.mongodb.net/admin
Connect

Click the **Connect** button.

You can now browse thru the databases in MongoDB.

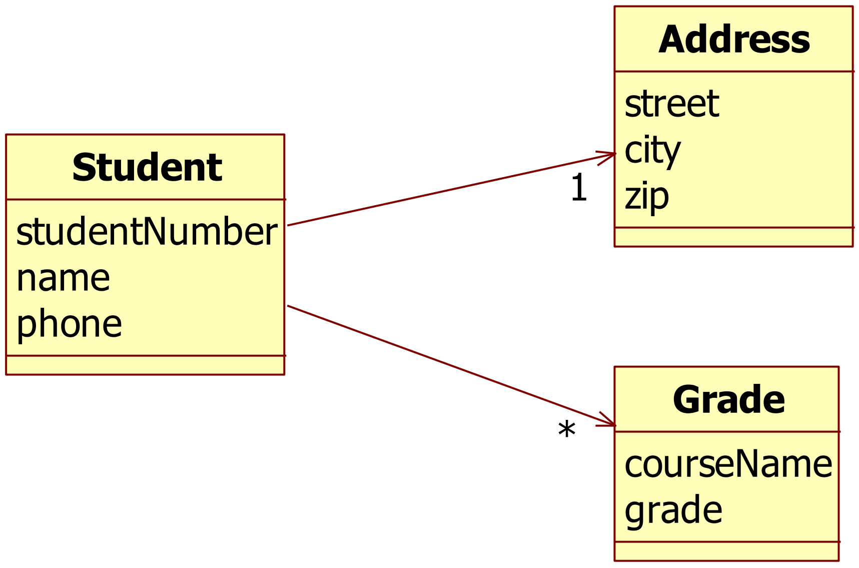
Modify the given project **Lesson8SpringBootMongoExample** so that we have the following domain classes:



Create a few students and save them in MongoDB.
Then write the following queries:

- **Find the Students with a certain name**
- **Find the Students with a certain phone number**
- **Find the Students from a certain city using @query**

Then modify the application to the following domain model:



Then write the following queries:

- **Find the Students that took a certain course with a given name**
- **Find the Students with an A+ for a certain course name**

Part C:

Write an application using Spring JPA that stores 10.000 Persons in the MySQL database where every person has a list of 10 Pet objects. A Person has a first name and last name and a Pet has a name and an age. Write a loop that creates all Person and Pet objects

After storing the 10.000 Persons with their Pets, retrieve all 10.000 Person with their Pets.

Measure the time it took to store all the Person and the time it took to retrieve all the Persons.

Then write a new application that does exactly the same, only it uses the MongoDB instead of MySQL. Then compare the times it took for both inserting the data and retrieving the data for the different databases.

Part D:

Rewrite the Bank application from Part A so that all data is stored in MongoDB and not in MySQL.

What to hand in:

1. A separate zip file with the solution of part A
2. A separate zip file with the solution of part B
3. A separate zip file with the solution of part C
4. A separate zip file with the solution of part D