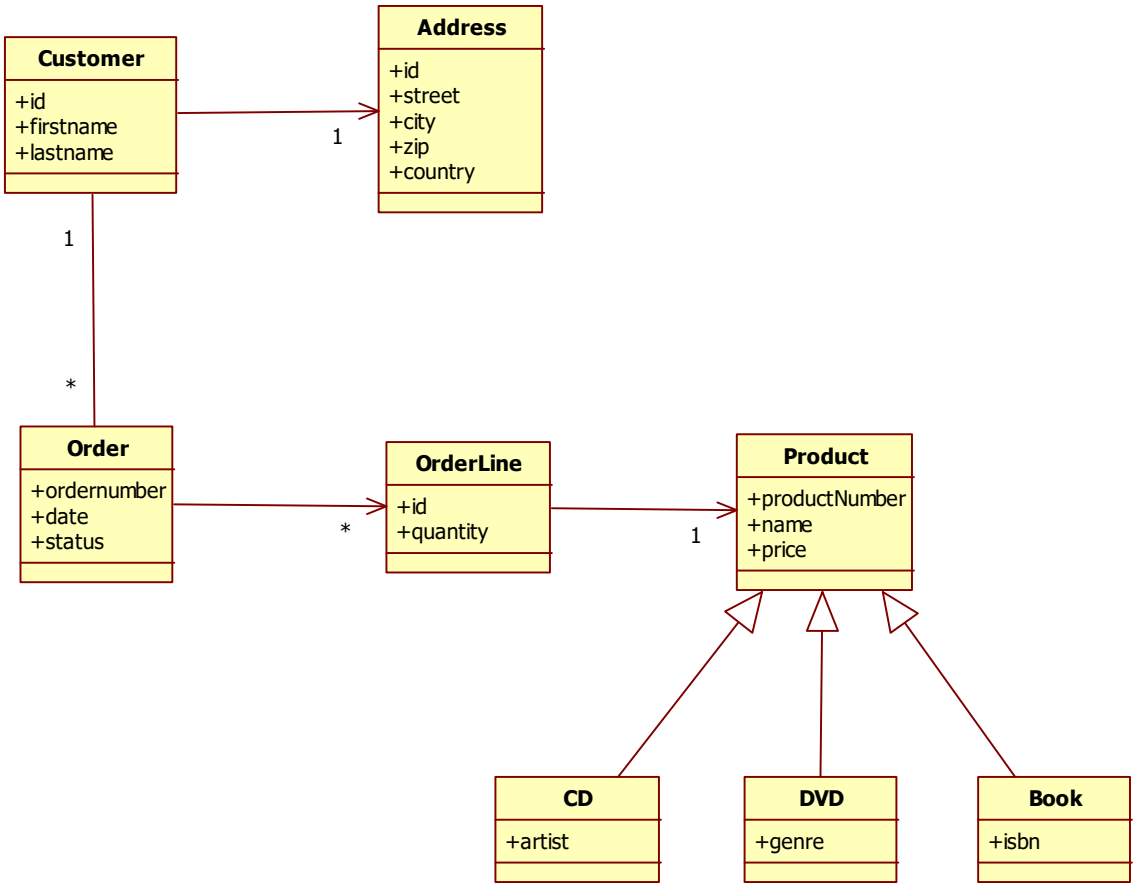
Lab 6

Part A:



In the solution of lab 5 part A, write the following queries and see if they work correctly:

Write queries using method names for the following queries.

- Give all customers.
- Give all CD's from U2 with a price smaller than 10 euro
- Give all customers with zip code 2389HJ
- Give all customers who ordered a DVD with the name Rocky3

Write a named query for the following queries:

- Give all customers from the USA.
- Give all CD's from a certain artist

Write a JPQL query with @Query for the following queries:

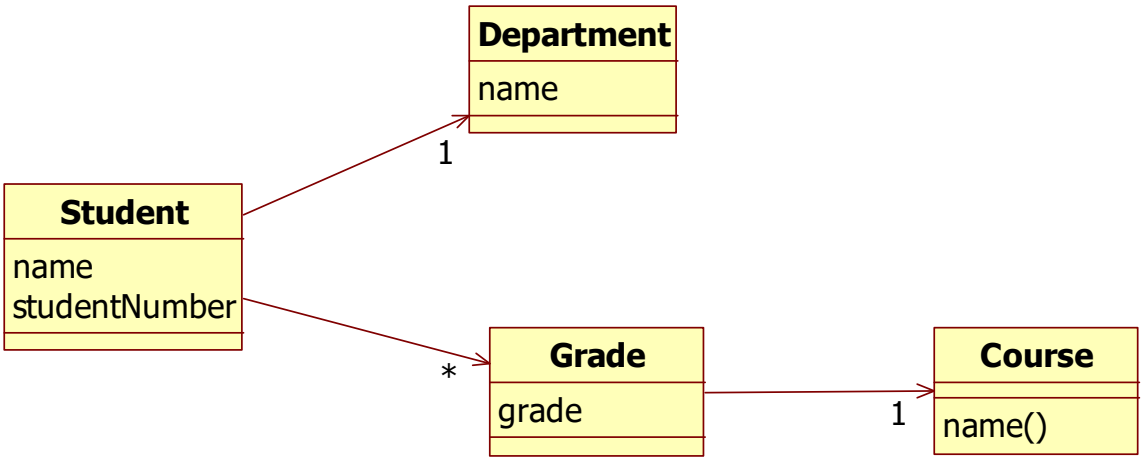
- Give the ordernumbers of all orders with status 'closed'
- Give the first and lastnames of all customers who live in Amsterdam.
- Give the ordernumbers of all orders from customers who live in a certain city (city is a parameter).
- Give all CD's from a certain artist with a price bigger than a certain amount (artist and amount are parameter2).

Write a native query for the following queries:

- Give all addresses in Amsterdam.
- Give all CD's from U2.

Part B

Create the following entities and store them with JPA in the database:



Then perform the following queries:

- Get all students from a certain department
- Get all students who took a course with a certain name.

Make sure that you optimize the mapping and the queries so that we get minimal database access

Part C

Given is the project **Lab6PartC**.

The application first inserts 10.000 Books in the database of an Library.

Then the code will update the locationCode of every book.

Then the code will remove all old books which are published before 1950.

When we run the code, the application measures how long it takes to do these actions:

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.5.4)

Changing the location code of all books took 10688 ms
Removing old books took 5673 ms

Process finished with exit code 0
```

As you can see, both operations took a really long time.

The exercise is to optimize the performance of the code for changing the location code and for removing old books. Both operations should take not more that 100 ms.

Part D:

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.

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