Lab 4

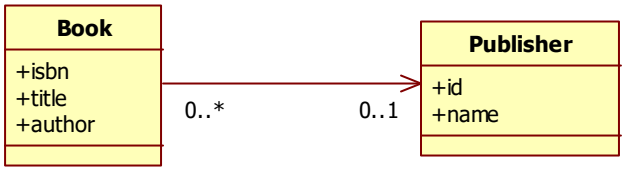
Part A:

Write the following entities and store them in the database. Then retrieve the first class from the database and check if the 2nd class(es) are also retrieved from the database.

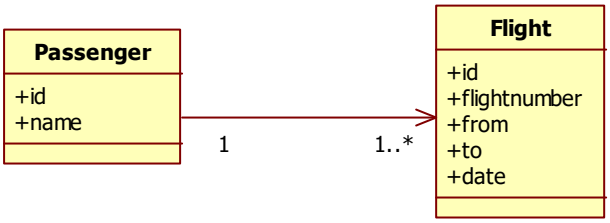
a) Create a **Bidirectional OneToMany** association between **Department** and **Employee**.

So in this case you save the Department in the database and make sure the corresponding Employee is also saved at the same time.
Then you retrieve the Department from the database and check if the Employee is also automatically retrieved.

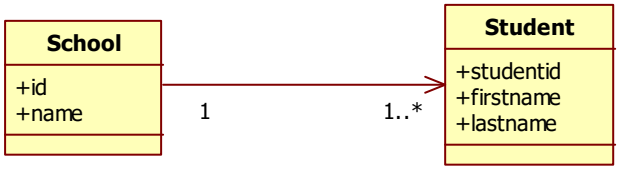
b) Create an **Optional Unidirectional ManyToOne** association between **Book** and **Publisher** without using NULL fields in the database



c) Create a **Unidirectional OneToMany** association between **Passenger** and **Flight** using a **List.** We need to keep the order of the flights exactly the same as the order we added the flights. Note that **from** and **to** are both reserved keywords and should be mapped to a different column name.

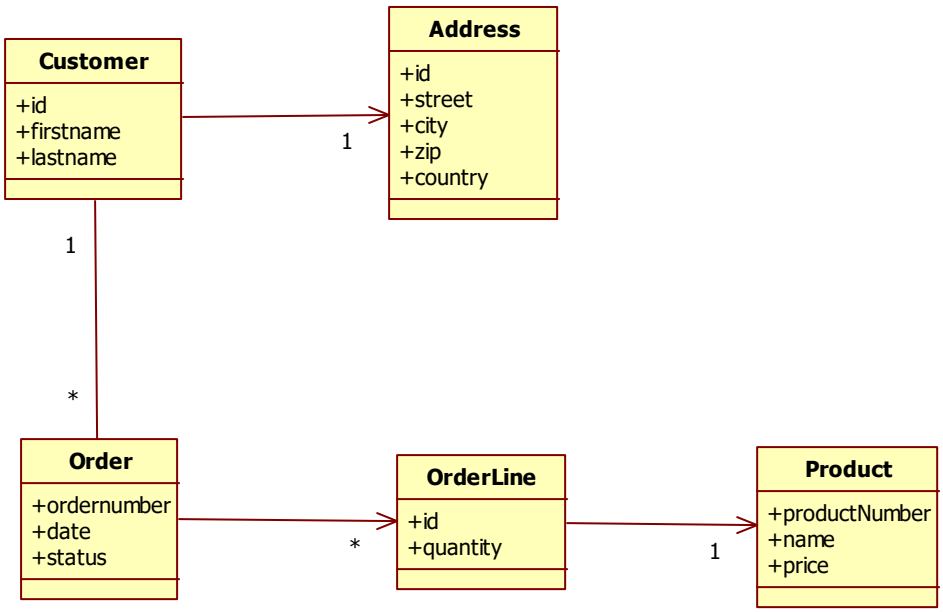


d) Create a **Unidirectional OneToMany** association between **School** and **Student** using a **Map**, where **studentId** is used as the key for the map.



Part B

Write the following entities and store them in the database:



Please note: "Order" is a SQL reserved keyword; you can use the @Table annotation to provide a different table name.

Test the application by saving an order to the database, and all related classes should also be saved to the database automatically.

Then retrieving the order from the database and check if all the other objects are also automatically retrieved.

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