

# **DAM. UNIT 3. ACCESS USING OBJECT- RELATIONAL MAPPING (ORM). HIBERNATE PART 2. NON ASSESSABLE EXERCISES**

# **DAM. Acceso a Datos (ADA) (a distancia en inglés)**

## **Unit 3. ACCESS USING OBJECT-RELATIONAL MAPPING (ORM)**

### **Part 2. Access using Hibernate annotations. Non assessable exercises**

**Abelardo Martínez**

Based and modified from Sergio Badal ([www.sergiobadal.com](http://www.sergiobadal.com))

**Year 2024-2025**

## Aspects to bear in mind

### Important

**If you look for the solutions surfing the Internet or asking the oracle of ChatGPT you will be fooling yourself.** Keep in mind that **ChatGPT is not infallible or all-powerful.**

It is a great tool to speed up your work once you have mastered a subject, but using it as a shortcut when acquiring basic skills and knowledge seriously undermines your learning. If you use it to get solutions or advice on your own, check the proposed solutions carefully as well. Try to solve the activities using the resources we have seen and the extended documentation you will find in the "Virtual Classroom".

## Tips for programming

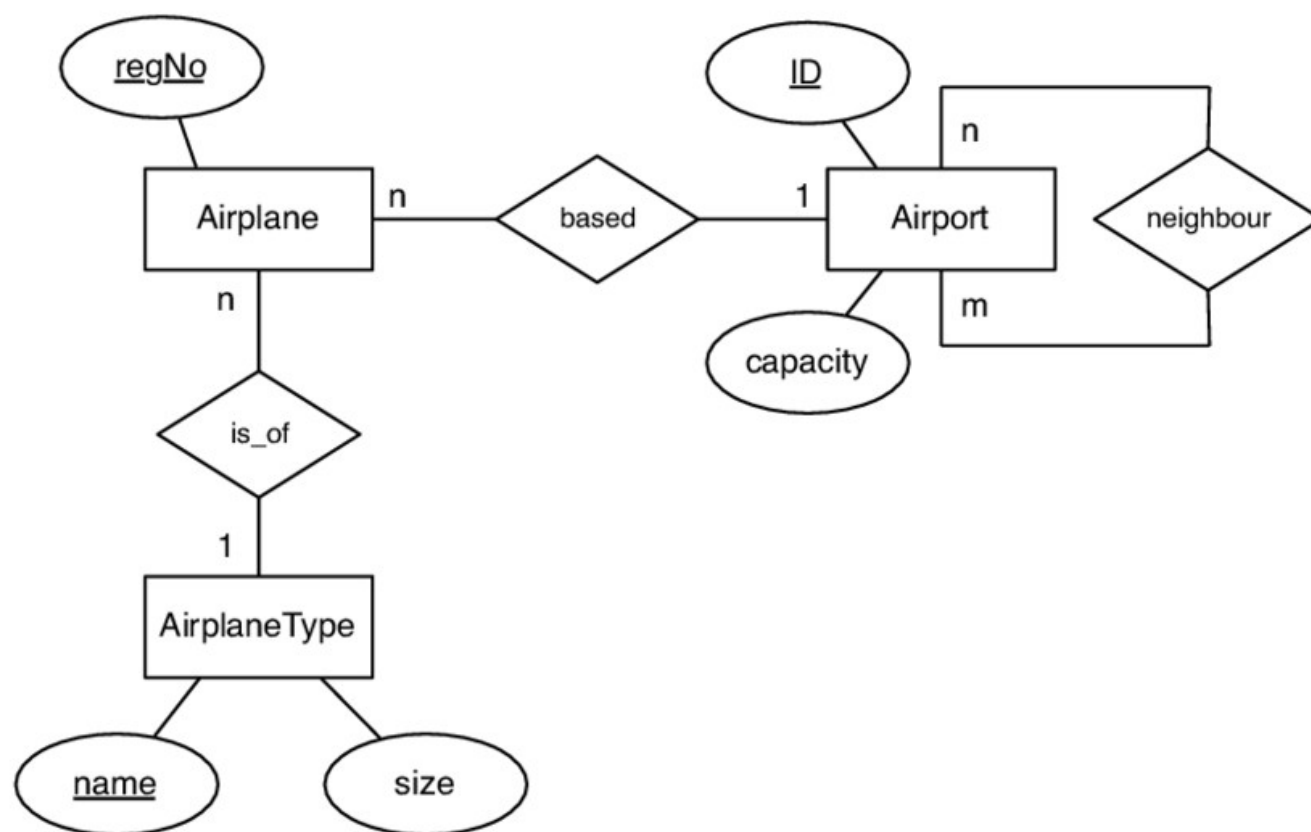
**We advice to follow the next coding standards:**

- One instruction per line.
- Add comments to make your code clearer and more readable.
- Use the Hungarian notation to recognise the type of variables at first sight.
- If necessary, we strongly recommend using buffer-based solutions.
- Remember that there are several ways to implement a solution, so choose the one you like best.

# 1. Console mode. Managing a MySQL relational database with Hibernate annotations

## Activity (non assessable)

Create a Java project to manage this E-R diagram in **MySQL** using **Maven** and **Hibernate ANNOTATIONS**.



**ATTENTION:** Use the proper exceptions when accessing to databases via Hibernate.

You MUST work with **Java**, **Maven** and **Hibernate**, using the new style as explained this week (mapping classes and using **annotations**). **You can reuse code from the previous task with Hibernate classic.**

Once you have the framework ready, run these operations:

- “Insert 3 aeroplane types, 3 airports and 3 aeroplanes, using all relationships”
- “Set a neighbourhood between the two first airports”
- “List all aeroplanes with its relationships”
- “List all airports with its relationships”
- “Remove all airports”
- “Remove all aeroplanes”
- “Remove all aeroplane types”



Licensed under the [Creative Commons Attribution Share Alike License 4.0](https://creativecommons.org/licenses/by-sa/4.0/)