

2nd December 2019

At this point Alice is looking for an application to easily manage the weapons and sales contacts. You offer her a very old but efficient tool for this type of work, a Database. She thinks a 3D holographic base would be more appropriate, but trusts your work.

First, you design the data model: In general terms, there are factions, with the name as PK, the contact person, the planet, the number of controlled systems, and the date of the last purchase. The factions have weapons to sell to other factions. The weapons have a price, a name and a unique id as PK. To keep track of the sales, the number of units, and the date of sale will be stored.

```
classDiagram
    class spies {
        id INT(11)
        sname VARCHAR(200)
        srace VARCHAR(200)
    }
    class battles {
        id INT(11)
        bname VARCHAR(200)
        faction_one VARCHAR(200)
        faction_two VARCHAR(200)
        bplace VARCHAR(200)
        bdate DATE
        id_spy INT(11)
    }
    class faction {
        fname VARCHAR(200)
        contact VARCHAR(200)
        planet VARCHAR(200)
        number_controlled_systems INT(11)
        date_last_purchase DATE
    }
    class weapons {
        id INT(11)
        wname VARCHAR(200)
        wprice DOUBLE
    }
    class sales {
        id INT(11)
        id_weapons_faction INT(11)
        units INT(11)
        sldate DATE
    }
    class weapons_factions {
        id INT(11)
        name_faction VARCHAR(200)
        id_weapon INT(11)
    }

    spies "1" -- "1..*" battles
    battles "1..*" -- "1..*" faction
    faction "1" -- "1..*" weapons_factions
    weapons_factions "1..*" -- "1..*" weapons
    weapons_factions "1..*" -- "1" sales
    sales "1..*" -- "1..*" weapons_factions
```

JDBC:

0. (1) Load initial data in the database from XML, a code that was used in the time of the Great Creator, when there was only one Light. You can find the XML file within the exam documents.

1. (0.75) Alice's cousin damaged his brain in a battle. To keep him busy, he is in charge of recording weapons of the rebels; to avoid mistakes, the program will delete the insertion immediately.

2. (0.75) Insert a new battle, inserting faction if it does not exist.

3. (0.75) Update the price of a weapon

4. (0.5) Select all factions, and all weapons

5. (0.5) Select the name and the price of all the weapons of a faction.

6. (0.75) Delete the data of a spy and his/her battles, so that if something happens to them, they can deny their existence.

Spring template:

7 (0.75) Select the data of all weapons purchased by a faction from a given date.

8 (0.75) Select all battles reported by a spy, returning battle name and spy name

9 (1) Update the name of a faction, a frequent task when the boss is killed and another takes control.

10 (0.75) Delete all weapons that are not sold

11 (0.75) Delete all data of a faction.