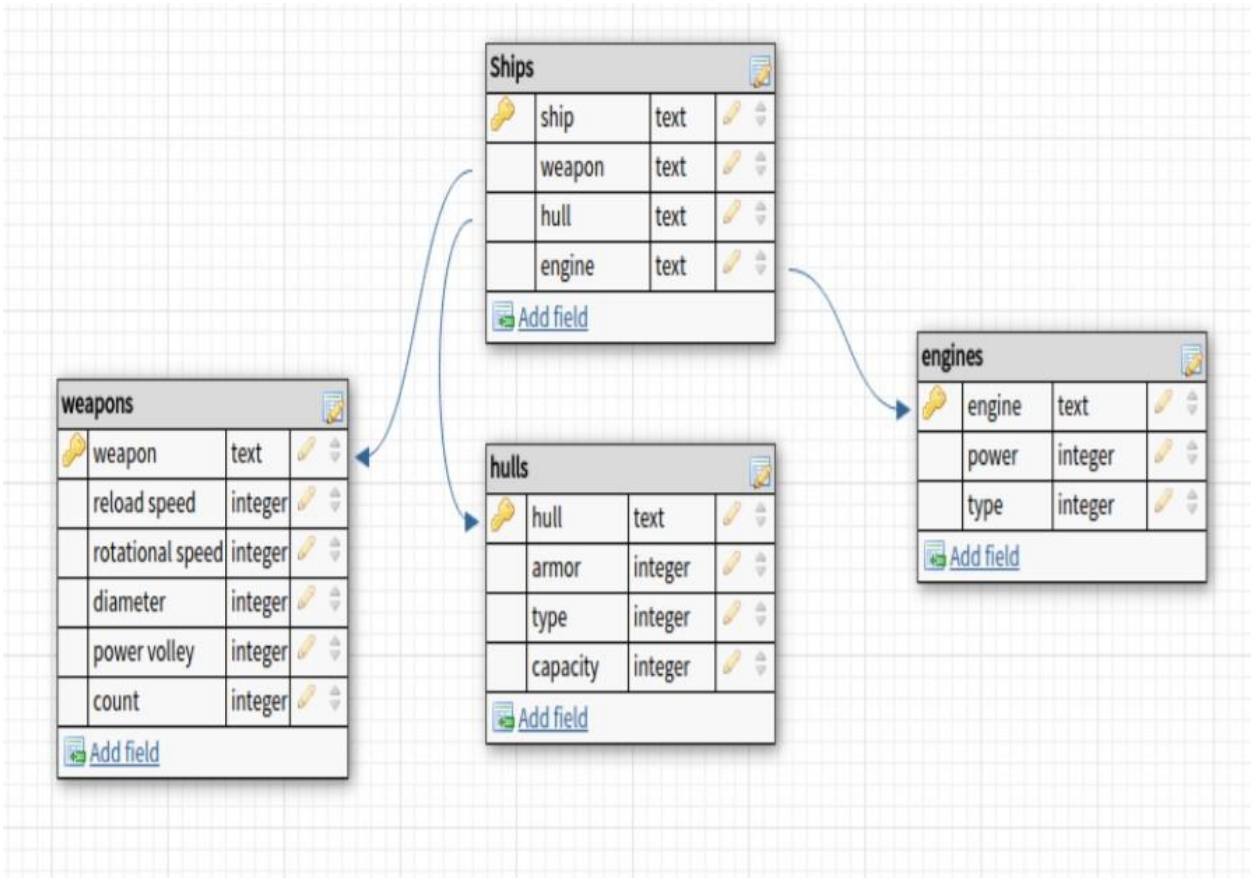
Test task for a test QA automation specialist

The assignment is designed to assess the work with Python, the level of proficiency in programming culture. To show the candidate what they will have to deal with. Despite this, the data in the assignment has a very simplified scheme relative to the real one.

Objectives of the task

1. Write a python script that creates an SQLite database according to the specified scheme.
   Primary key - text field weapon / ship / hull / engine respectively.



2. Create a script that will randomly fill values in the created database.
   The names: Ship-1, Ship-2, Weapon-1, etc. are quite suitable.

   The number of records for each table:
   ships: **200**
   weapons: **20**
   hulls: **5**
   engines: **6**

   Value range for integer parameters: **1-20**

3. Develop a session-scope fixture that gets the current state of the database and creates a temporary new database where the values are randomised:
    a. For each ship, **one** of the components is changed to a random one: hull, gun, or engine.
    OR
    b. Each component changes one of the randomly selected parameters to a random value from the allowable range (see above).

4. Implement automated tests that compare the data from the original database with the resulting randomized data:
    a. There should be three tests for each ship, checking its gun, hull and engine.
    b. The test should fall with **assert**:
        i. When the value of a component parameter does not match its pre-randomizer.
        Output example:

```
Ship-2, weapon-1
        reload speed: expected 1, was 2
Ship-2, hull-3
        type: expected 1, was 2
Ship-3, engine-6
        power: expected 22, was 13
```

        ii. When the gun, hull, or engine is changed.
        Output both the previous and current content.

```
Ship-5, engine-4
        expected engine-1, was engine-4
```

Requirements for the completed task:

- The version of Python interpreter – 3.8
- Tests should be implemented using the **pytest** framework
- Use **pytest.mark.parametrize** or **the pytest_generate_tests hook** as parameterization
- The run should result in 600 tests.
- The assignment should result in at least the following:
    - A script that creates and fills the initial database, a Python module that contains the tests.
    - (Optional) conftest.py module, contains fixtures and hooks.
- Code style – PEP8.