

Home Task for the **World of Warships, PC**

Position: Automation QA Engineer

Timeframe for completion: 7 days

Not restricted by NDA



Hello!

We truly appreciate your willingness to take on our Home Task.

At Wargaming, we believe that speed and high-quality work are our key advantages. That's why we are looking for people who can deliver strong results within a reasonable timeframe. Just like in real production.

Please note: we do not use the results of the Home Task anywhere beyond the evaluation process.

If you have any questions, please feel free to reach out to your Recruiter.

Good luck!



Home Task goal

The Home Task 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.

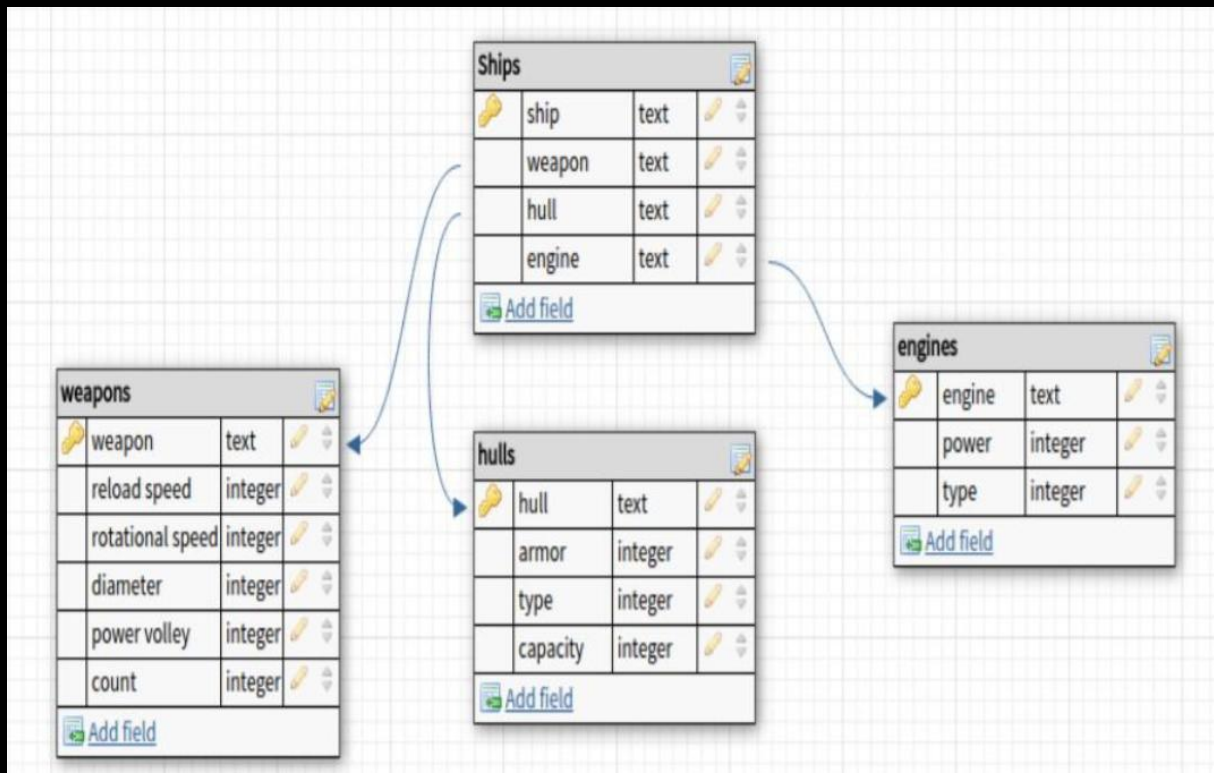
Requirements for the completed task

- 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 Home Task 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



I TASK

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



II TASK

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**



III TASK

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).



IV TASK

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:
 - B.1 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
```

- B.2 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
```

